

Titre: Optimisation de la conception d'un réseau de collecte d'énergie
Title: éolienne par recherche locale

Auteur: Guillaume Blanchet
Author:

Date: 2013

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Blanchet, G. (2013). Optimisation de la conception d'un réseau de collecte
Citation: d'énergie éolienne par recherche locale [Mémoire de maîtrise, École
Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/1176/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/1176/>
PolyPublie URL:

**Directeurs de
recherche:** Alain Hertz
Advisors:

Programme: Mathématiques appliquées
Program:

UNIVERSITÉ DE MONTRÉAL

OPTIMISATION DE LA CONCEPTION D'UN RÉSEAU DE COLLECTE D'ÉNERGIE
ÉOLIENNE PAR RECHERCHE LOCALE

GUILLAUME BLANCHET
DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(MATHÉMATIQUES APPLIQUÉES)
JUILLET 2013

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

OPTIMISATION DE LA CONCEPTION D'UN RÉSEAU DE COLLECTE D'ÉNERGIE
ÉOLIENNE PAR RECHERCHE LOCALE

présenté par : BLANCHET Guillaume

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

Mme LAHRICHI Nadia, Ph.D., présidente

M. HERTZ Alain, Doct. ès Sc., membre et directeur de recherche

Mme MARCOTTE Odile, Ph.D., membre

REMERCIEMENTS

Je remercie Alain Hertz de m'avoir dirigé et accepté comme disciple. Je remercie également Odile Marcotte, membre du jury, et Asma Mdimagh, étudiante au doctorat, pour leur aide et leurs conseils. Sans l'impératif monétaire, c'est avec enthousiasme que j'aurais poursuivi avec eux nos recherches au doctorat.

Je remercie le gouvernement du Québec d'avoir financé en partie mes études. C'est avec plaisir que je m'acquitterai désormais de mes impôts pour payer au suivant.

Je remercie mes parents ainsi que tous mes professeurs en physique et en math qui ont suscité ma curiosité pour la science, un royaume où il fait bon vivre et où l'optimisme est éclatant.

RÉSUMÉ

Le problème de l'optimisation de la conception de réseaux, plus particulièrement d'un parc éolien, intervient lorsqu'une compagnie souhaite louer une partie des câbles de transmission d'un réseau appartenant à un tiers parti (l'État ou une autre compagnie) et construire/acheter l'autre partie manquante.

Il est possible de reformuler ce problème comme le problème de l'arbre de Steiner en adaptant la fonction de coût des câbles de transmission.

Une fois cette adaptation faite, une heuristique employant une méthode exacte est développée.

L'heuristique développée se découpe en deux pans :

1. la construction d'une solution réalisable initiale itérativement en utilisant un algorithme de plus court chemin, et
2. l'amélioration de cette solution par une série de déconnexions de certaines parties de la solution et de leur reconnexion à l'autre partie restante non déconnectée de la solution en question.

Les résultats montrent que la méthode de construction de la solution initiale échoue parfois à trouver une solution réalisable. Ils montrent également que pour de larges exemplaires, le prototype d'amélioration de la solution initiale s'avère très efficace et pourra vraisemblablement s'attaquer à de très gros problèmes, impensables à résoudre avec les méthodes actuelles, si quelques ajustements sont faits.

ABSTRACT

The problem of optimizing the design of network, in particular wind farm network, occurs when a company wishes to rent cables from a part of an existing transmission network owned by a third party (the government or another company) and construct or buy cables from the other part missing.

It is possible to reformulate this problem as the Steiner tree problem in graph by adjusting the cost function of the transmission cables.

Once this adaptation made, a heuristic using an exact method is developed.

The heuristic developed is divided into two sections:

1. the construction of an initial feasible solution iteratively using the shortest path algorithm, and
2. improving this solution by a series of disconnection of parts of the solution and their connection to the other remaining portion not disconnected from the solution at hand.

The results show that the method of construction of the initial solution sometimes fails to find a feasible solution. They also show that for large instances, the heuristic is very effective and will probably find good solutions also for very large problems, unthinkable to solve with current methods, if some adjustments are made.

TABLE DES MATIÈRES

REMERCIEMENTS	iii
RÉSUMÉ	iv
ABSTRACT	v
TABLE DES MATIÈRES	vi
LISTE DES TABLEAUX	viii
LISTE DES FIGURES	ix
CHAPITRE 1 INTRODUCTION	1
1.1 Le problème de la conception	1
1.2 Plan du mémoire	3
CHAPITRE 2 REVUE DE LITTÉRATURE	5
CHAPITRE 3 FORMULATION	8
3.1 La formulation du réseau sous-jacent de Hertz <i>et al.</i> (2012)	8
3.2 Le programme linéaire de Hertz <i>et al.</i> (2012)	9
3.3 La formulation dans le graphe adjoint	11
3.3.1 Les coûts, les capacités et les contraintes	11
3.3.2 Quelques définitions	12
CHAPITRE 4 RÉOLUTION	17
4.0.3 La construction d'une solution initiale réalisable	17
4.0.4 Amélioration de la solution initiale	24
CHAPITRE 5 RÉSULTATS	35
5.1 Analyse des résultats	36
CHAPITRE 6 CONCLUSION	39
6.1 Synthèse des travaux	39
6.2 Limitations de la solution proposée	39
6.3 Améliorations futures	39

RÉFÉRENCES	41
----------------------	----

LISTE DES TABLEAUX

Tableau 5.1	Gains de performance résultant de l'agrégation des groupes de turbines et du processus de réparation	35
Tableau 5.2	Données des exemplaires et résultat de Hertz <i>et al.</i> (2012)	36
Tableau 5.3	Résultats	36
Tableau 5.4	Résultats avec temps limite = 30 secondes	37

LISTE DES FIGURES

Figure 1.1	Exemple de parc éolien.	3
Figure 1.2	Une solution de la figure 1.1.	4
Figure 3.1	Une solution de la figure 1.2 dans le graphe adjoint.	14
Figure 3.2	La solution optimale de coût 30 dans le graphe original.	15
Figure 3.3	L'anti-arborescence optimale dans le graphe adjoint.	16
Figure 4.1	Il est clair qu'il est impossible de connecter la turbine 6 si toutes les turbines 3,4,2,5 évitent de passer par (9, 1) dans l'ordre de connexion 3,4,2,5,8,7,6 dans l'algorithme 1. Par contre, si nous avons choisi l'ordre 6,7,8,5,3,4,2 pour nos plus courts chemins dans l'algorithme 1, la solution résultante serait réalisable.	19
Figure 4.2	Résolution par CPLEX du modèle 3.2 pour $T = \{6, 7, 8\}$	21
Figure 4.3	Le graphe de la figure 1.1 après agrégation.	22
Figure 4.4	23
Figure 4.5	24
Figure 4.6	24
Figure 4.7	La première étape consiste à recevoir le graphe et son ensemble de turbines.	28
Figure 4.8	Ensuite, il faut agréger les groupes.	29
Figure 4.9	Puis, il faut générer la solution initiale grâce à l'algorithme 2. L'ordre dans lequel les turbines ont été connectées avec l'algorithme de plus court chemin est comme suit : 20,5,2,21,6. C'est une solution réalisable, il est inutile de tenter d'autres ordres.	30
Figure 4.10	Vu qu'ici $\alpha = 1$, une seule itération de l'algorithme est opérée. Il est supposé ici que $\beta = 2$ et que les 2 turbines à déconnecter sont 20 et 21. Le lien (15,12,1) est désormais interdit ; il faut le traiter comme un lien (15,12,0) en décrémentant son numéro parallèle.	31
Figure 4.11	Le lien (15,12,1) a été remplacé par le lien (15,12,0).	32
Figure 4.12	Les turbines 20 et 21 sont reconnectées de façon optimale grâce au modèle de Hertz <i>et al.</i> (2012).	33

Figure 4.13 Le graphe est désagrégé. Chaque groupe de turbines est déconnecté et reconnecté par l'algorithme de Hertz *et al.* (2012) jusqu'à ce que la solution ne s'améliore plus (ici, cette procédure est inutile, car la connexions construite *a priori* des groupes était la meilleure pour la solution finale). 34

CHAPITRE 1

INTRODUCTION

Le problème de l'optimisation de la conception d'un parc éolien intervient lorsqu'une compagnie souhaite acheminer à travers un réseau partiellement existant vers une sous-station de transmission le courant généré par ses éoliennes. Elle doit alors acheter des câbles pour relier ces éoliennes au réseau électrique existant, puis elle doit louer l'utilisation des lignes de transmission dont elle a besoin pour connecter ses éoliennes jusqu'à la sous-station.

C'est un problème très concret consistant simplement à minimiser le coût total regroupant les achats et locations de câbles tout en acheminant le courant généré vers la sous-station de transmission. La fonction de coût est déterminée par le tiers qui possède le réseau électrique (souvent l'État) dans le cas de la location des lignes et dans le cas de l'installation/construction des câbles, il faut tenir compte, entre autres, du prix des matériaux de la main-d'œuvre et d'un amortissement sur les années d'utilisation.

Les exemplaires de réseaux traités dans ce mémoire sont fournis par la compagnie de génie-conseil Hatch dont les clients doivent résoudre ce genre de problème.

Déjà, des modèles exacts de résolution ont été développés (Hertz *et al.* (2012)) et fonctionnent bien. Cependant, pour de grands parcs éoliens (des parcs qui possèdent au-dessus de 50 éoliennes généralement), le temps de calcul devient vite inacceptable. Une approche heuristique serait donc utile pour traiter les plus gros problèmes. L'objectif de ce mémoire est de développer cette approche heuristique.

1.1 Le problème de la conception

Un parc éolien est formé de 5 composantes essentielles :

1. de turbines éoliennes qui servent à générer le courant ;
2. de câbles qui servent à acheminer ce courant vers des relais (ces câbles font partie du réseau *sous-terre*) ;
3. de relais servant à connecter le réseau *sous-terre* au réseau *hors-terre* ;
4. de lignes de transmission formant le réseau *hors-terre* ;
5. de la sous-station vers où tout le courant généré doit être acheminé.

L'utilisation de plusieurs câbles ou lignes parallèles est possible. Parfois, l'utilisation d'un lien (câble ou ligne) parallèle supplémentaire profite d'un coût dégressif, c'est-à-dire que le

prix par lien d'utiliser un seul lien sera plus élevé que celui d'en utiliser deux, par exemple.

Le courant ne peut circuler dans deux sens différents sur un même câble ou sur une même ligne.

Il faut préciser ce que signifient *utilisation* et *capacité* d'un câble ou d'une ligne. Dans le problème de la conception, il est supposé que chaque turbine produit une unité de flot électrique qui circulera dans le réseau. L'utilisation correspond au nombre d'unités de flot qui circulent sur un câble ou une ligne, et la capacité à l'utilisation maximale permise sur ce lien. Un lien a un coût fixe et donc il ne coûte pas plus cher de l'utiliser jusqu'à sa capacité que de l'utiliser pour une seule unité de flot. Cette particularité fait en sorte que la fonction de coût des liens dépend de la solution courante, une différence majeure avec des problèmes plus classiques apparentés comme celui de l'arbre de Steiner qui sera discuté dans la revue de littérature.

Une contrainte supplémentaire apporte son lot d'originalité : le courant dans le problème de la conception ne doit pas *bifurquer*, c'est-à-dire que lorsque deux unités de flot se rencontrent sur un câble ou une ligne, elles ne peuvent plus se séparer et emprunter des chemins différents dans le réseau.

On donne un exemple de réseau dans les figures 1.1. Sur la figure, les turbines sont représentées par les triangles, les jonctions du réseau par les cercles en traits pleins, les relais par les cercles en traits pointillés et la sous-station par le carré. Les pointillés signifient que les composantes appartiennent au réseau sous-terre. Une capacité maximale de 2 unités a été fixée sur chaque câble. Les câbles noirs coûtent 2\$ et les câbles gris coûtent 1\$. On ne peut utiliser un câble gris que si on a utilisé auparavant le câble noir qui lui est parallèle. Cette structure est un petit exemple très réduit imitant l'allure des véritables parcs éoliens plus gros rencontrés. Le réseau hors-terre ressemble, en effet, souvent à ce genre de schéma quadrillé, car ses lignes suivent le réseau routier.

On explicite une solution sur la figure 1.2. Les arcs non utilisés ont été enlevés pour mieux voir la structure de la solution. Mentionnons la présence d'un cycle défini par les noeuds 1,9,10,11,12,13. Cette solution coûte 34\$. Elle n'est pas la solution optimale. Les chiffres sur les arcs représentent le flot sur chaque arc.

Ce mémoire apporte une technique efficace pour visiter des solutions intéressantes du problème de l'optimisation de la collecte d'énergie dans un parc éolien. Cette technique se base sur des connexions et des déconnexions successives de turbines de la solution courante.

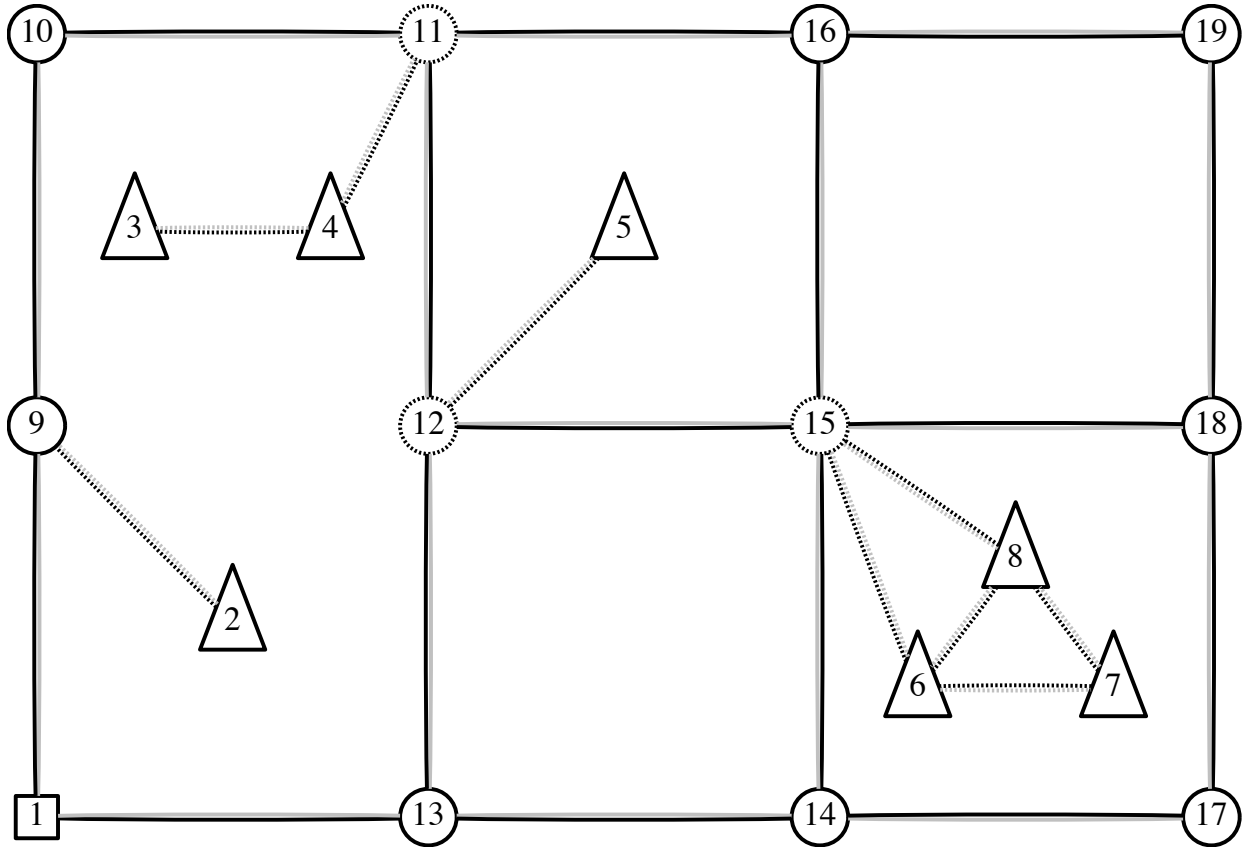


Figure 1.1 Exemple de parc éolien.

1.2 Plan du mémoire

D'abord, au chapitre 2, une revue de la littérature concernant principalement le problème de l'arbre de Steiner sera présentée. Ce problème servira à guider la formulation du problème de l'optimisation de la conception d'un réseau de collecte d'énergie éolienne présenté au chapitre 3. Puis, au chapitre 4, une résolution du problème est proposée. Elle résout un réseau prétraité où des noeuds ont été fusionnés selon des critères géographiques pour accélérer le temps de calcul. Enfin, les résultats sont présentés au chapitre 5 avec une analyse qui permet de transformer le prototype de la résolution en un véritable algorithme industriel capable de s'attaquer aux plus gros problèmes d'optimisation de la conception de réseau.

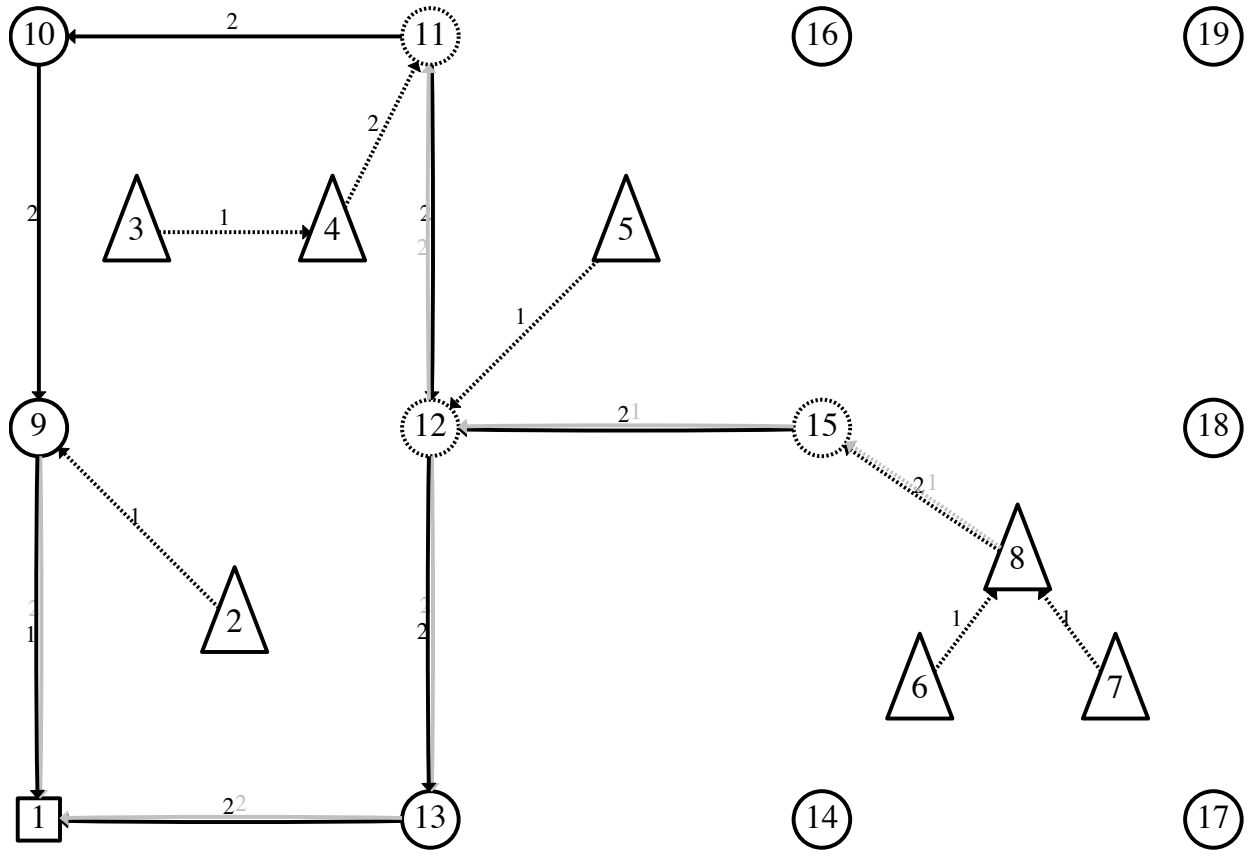


Figure 1.2 Une solution de la figure 1.1.

CHAPITRE 2

REVUE DE LITTÉRATURE

L'optimisation de la conception de la collecte d'énergie dans un parc éolien tel qu'elle se présente pour la compagnie Hatch est plutôt singulière. Mis à part l'article sur les méthodes exactes par Hertz *et al.* (2012), mentionnons l'article de Berzan *et al.* (2011), où les auteurs considèrent un problème plus simple qu'ils divisent en trois niveaux ou trois problèmes : un problème du circuit consistant à relier ensemble un groupe de turbines de façon minimale, un problème de la sous-station consistant à assigner des groupes de turbines à la sous-station de façon optimale, et un problème du parc entier consistant à la fois à déterminer non seulement les groupes de turbines, mais à déterminer à quelle sous-station il faut les assigner de façon optimale. Lorsqu'il y a un seul type de câble, les problèmes du circuit et de la sous-station se réduisent à celui (classique) de l'arbre de poids minimum en théorie des graphes. Cette modélisation compartimentée ne donne pas du tout de moyen de reconnecter la sous-station à l'ensemble de turbines de façon efficace pour Hatch. Il est difficile de savoir si la connexion des turbines, fait de façon indépendante, pourra être optimale, voir simplement réalisable lors de l'intégration de la sous-station dans le parc éolien, étant donné les capacités, les contraintes de non-bifurcations du courant et du sens du courant.

Fagerfjäll (2010) quant à lui s'est intéressé à deux modèles : un modèle de production et un modèle d'infrastructure. Le modèle d'infrastructure est celui relié au problème de Hatch. Cependant, les emplacements des turbines dans le modèle de Fagerfjäll (2010) ne sont pas connus complètement et l'auteur a testé son modèle pour des exemplaires de réseaux n'ayant pas plus de 30 turbines. Hatch est aux prises avec des exemplaires pouvant avoir plus de 100 turbines.

Un graphe non orienté est noté $G = (V, E)$, où V représente l'ensemble de sommets et E l'ensemble d'arêtes ou paires de sommets. $G = (V, A)$ est la notation pour un graphe orienté ayant A comme ensemble d'arcs ou couples de sommets. Une fonction de coût $c(e)$ peut être associée aux arêtes/arcs $e \in E, A$ d'un graphe.

Le problème classique de l'arbre de poids minimum dans un graphe $G = (V, E)$ en théorie des graphes consiste simplement à trouver le sous-graphe $H = (V, \mathcal{E})$, $\mathcal{E} \subseteq E$ tel qu'il existe un chemin entre deux noeuds quelconques $u, v \in V$ dans H et tel que $\sum_{e \in \mathcal{E}} c(e)$ est minimisée.

Le problème de Hatch est bien différent de celui de l'arbre de poids minimum pour toutes sortes de raisons, la principale étant qu'il faut non pas connecter tous les sommets entre eux, mais seulement un sous-ensemble de sommets, c'est-à-dire ceux regroupant les turbines et la

sous-station.

Le problème consistant à trouver un arbre de poids minimum entre un sous-ensemble de sommets se nomme le problème de l'arbre de Steiner en théorie des graphes et est \mathcal{NP} -dur, Karp (2010). On peut retrouver sa formulation dans Hwang et Richards (1992) ou Maculan (1987), tant sous sa forme non orientée qu'orientée.

Plus précisément, si on reprend la définition de Maculan (1987), soit $G = (V, A)$ un graphe orienté dont chaque arc $a \in A$ est muni d'un coût $c(a) \geq 0$. L'ensemble V reliant des noeuds est partitionné en trois sous-ensembles : $\{s\}$, T et L où s est la racine, T l'ensemble de noeuds obligatoires et R l'ensemble de noeuds facultatifs ou noeuds de Steiner.

Le problème de Steiner sur le graphe G orienté consiste à déterminer un sous-graphe partiel de G , $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ où $\{s\} \cup T \subseteq \mathcal{V} \subseteq V$ et $\mathcal{A} \subseteq A$ de coût total minimal, permettant de relier par un chemin s à chaque sommet de T en utilisant, éventuellement, un ou plusieurs sommets facultatifs.

Pour Hatch, le noeud racine consiste en une sous-station de transmission où doit être acheminé le courant des turbines qui sont les noeuds obligatoires correspondant à ce qui a été défini comme l'ensemble T dans l'arbre de Steiner. Les noeuds du réseau L sont simplement les liens du réseau intermédiaire loué à un tiers (ici, les liens sont des noeuds, car la résolution se fera dans le graphe adjoint).

Évidemment, la conception d'un parc éolien comporte certaines contraintes supplémentaires, comme celle sur la non-bifurcation du courant, ou celle sur la capacité des arcs.

Étant donné que le problème de l'arbre de Steiner est \mathcal{NP} -dur, il n'est pas raisonnable pour des exemplaires contenant un grand nombre de turbines de le résoudre avec des méthodes exactes. Aussi, Hertz *et al.* (2012) n'arrivent pas à résoudre certaines instances (généralement celles au-dessus de 50 turbines) du problème de Hatch en des temps raisonnables.

Plusieurs heuristiques ont été proposées pour résoudre le problème de l'arbre de Steiner, voir par exemple Chakraverty *et al.* (2006) et Takahashi et Matsuyama (1980). La plus utilisée semble la *Minimum Path Heuristic* (MPH) par Takahashi et Matsuyama (1980), consistant à générer une suite de plus courts chemins entre des paires de sommets de l'arbre de Steiner de façon à construire ce dernier. L'ordre dans lequel on prend les sommets de l'arbre pour générer la suite de plus courts chemins peut faire varier beaucoup la valeur de la solution obtenue. Étant donné cet inconvénient, Khoury et Pardalos (1996) ont suggéré une correction en associant des poids plus élevés à certaines paires de sommets dans la suite. Certains ont tenté des métaheuristiques. Des essais étonnants ont été faits avec des algorithmes génétiques, voir Esbensen (1994) et Kapsalis *et al.* (1993). Cependant, les meilleures performances computationnelles proviennent des méthodes basées sur la recherche tabou de Glover et Laguna (1998) (utilisant, entre autres, la MPH de Takahashi et Matsuyama (1980)). En effet,

Gendreau *et al.* (1998) sont arrivés à de très bons résultats sur les exemplaires standards de la *OR-Library*. Ribeiro et De Souza (2000) sont parvenus à raffiner cette dernière méthode encore davantage. Cette méthode consiste à définir le voisinage d'une solution comme étant l'ensemble des solutions pouvant être générées en retirant ou en ajoutant un noeud de Steiner (les noeuds qui ne sont ni la sous-station, ni les turbines dans notre cas) de la solution. Le retrait d'un noeud provoque l'apparition de sous-arbres qu'il faut reconnecter, ceci est fait grâce à l'algorithme de Kruskal ou de Prim. L'ajout d'un noeud, quant à lui, provoque l'apparition d'un cycle qu'il faut briser, ceci est fait en enlevant l'arête la plus coûteuse.

Il reste que l'approche des métaheuristiques pour résoudre le problème de l'arbre de Steiner ne tient pas compte de contraintes exotiques sur les capacités et sur les coûts d'arcs parallèles comme celles en jeu ici. Il faut adapter ces idées pour qu'elles reflètent les contraintes du problème de Hatch.

CHAPITRE 3

FORMULATION

Ce chapitre donne la formulation formelle du problème dans le graphe adjoint, formulation discutée de façon préliminaire dans l'introduction.

3.1 La formulation du réseau sous-jacent de Hertz *et al.* (2012)

La formulation de Hertz *et al.* (2012) consiste à décrire le graphe orienté $G = (V, A)$ du problème sous-jacent. Elle est reprise ici.

Le graphe doit être orienté puisque le courant circule dans une seule et même direction sur un câble ou une ligne de transmission. L'ensemble V est l'union disjointe de T (l'ensemble des turbines), R (l'ensemble des points terminaux des lignes de transmission) et $\{0, s\}$ (où s est la sous-station et 0 un noeud source). L'ensemble A est l'union disjointe des cinq catégories d'arcs suivantes :

1. tous les arcs de la forme $(0, u)$ pour $u \in T$;
2. tous les arcs de la forme (u, v) et (v, u) pour chaque câble, avec $u, v \in T$;
3. tous les arcs de la forme (u, v) où $u \in T$ et $v \in R$;
4. tous les arcs de la forme (u, v) et (v, u) pour chaque ligne de transmission, avec $u, v \in R$;
et
5. tous les arcs de la forme (v, s) , où $v \in R$ et où il y a une ligne de transmission entre v et s .

Soit m le nombre maximal de liens parallèles entre deux noeuds u et v . Par exemple, il peut y avoir jusqu'à m câbles entre les turbines u et v . Les liens sont notés $(u, v, 1)$, $(v, u, 1)$, $(u, v, 2)$, etc. mais seulement un parmi (u, v, k) et (v, u, k) est sélectionné pour chaque k (le courant ne peut circuler que dans une direction sur un même lien). La valeur de m est égale à 4 pour Hatch.

La capacité d'un arc de catégorie 1 est 1, car une unité de courant est produite par chaque turbine. La capacité d'un arc de catégorie 2 ou 3 est égale à C_{ug} (*ug* pour *underground*), une constante donnée pour chaque exemplaire de réseau. La capacité d'un arc de catégorie 4 ou 5 est égale à C_{ag} (*ag* pour *above ground*), une constante donnée pour chaque exemplaire de réseau.

On note c_{uv}^k le coût du k -ième lien entre le noeud u et le noeud v . Le coût de chaque lien additionnel entre deux noeuds u et v est au plus le coût du lien parallèle précédent entre u et v . Donc si $c_{u,v}^k$ représente le coût du lien (u, v, k) , la relation $c_{uv}^1 \geq c_{uv}^2 \geq \dots \geq c_{uv}^m$ est valable pour tout arc (u, v) (il faut noter que c_{uv}^k est égale à c_{vu}^k pour tout k). Le coût c_{0u}^k égale 0 pour tout k et tout $u \in T$, car le noeud source est fictif.

Le but du modèle est de minimiser le coût total des liens utilisés pour collecter le courant produit et l'acheminer à la sous-station.

3.2 Le programme linéaire de Hertz *et al.* (2012)

Comme mentionné dans l'introduction, Hertz *et al.* (2012) utilisent une méthode exacte pour résoudre le problème. Plus précisément, ils formulent un programme linéaire. Il est donné ci-bas, car il servira à résoudre des problèmes auxiliaires dans la résolution présentée au chapitre suivant.

Mais avant, il faut donner quelques définitions.

Soit A_1 l'ensemble des arcs (u, v) tel que $(v, u) \notin A$ et E l'ensemble des paires $\{u, v\}$ telles que $(u, v), (v, u) \in A$.

Soit t_{uv}^k , une variable de décision binaire égale à 1 si et seulement si (u, v, k) appartient à la solution et 0 autrement. Soit x_{uv}^k le flot circulant sur (u, v, k) . Comme mentionné plus haut, le flot ne doit pas bifurquer. Pour tenir compte de cette contrainte, la variable de décision binaire $y_{uvw}^{kk'}$ a été introduite. $y_{uvw}^{kk'}$ égale 1 si et seulement si tout le courant qui circule sur (u, v, k) circule aussi sur (v, w, k') . Enfin, $z_{uvw}^{kk'}$ représente la quantité de flot circulant sur (u, v, k) puis sur (v, w, k') .

$P(u)$ ($S(u)$) est l'ensemble des prédécesseurs (successeurs) de u , c'est-à-dire l'ensemble des noeuds v tel que $(v, u) \in A$ ($(u, v) \in A$).

La capacité de l'arc (u, v) est notée C_{uv} .

Le but du modèle est de minimiser le coût total des câbles et des lignes, c'est-à-dire $\sum_{(u,v) \in A} \sum_{k=1}^m c_{uv}^k t_{uv}^k$, où c_{uv}^k représente le coût du lien (u, v, k) .

L'ensemble K est l'ensemble des indices des arcs parallèles $(1, 2, \dots, m)$.

$$\min \sum_{(u,v) \in A} \sum_{k=1}^m c_{uv}^k t_{uv}^k \quad \text{s.t.} \quad (3.1)$$

$$\sum_{v \in P(u)} \sum_{k=1}^m x_{vu}^k - \sum_{v \in S(u)} \sum_{k=1}^m x_{uv}^k = \begin{cases} |T| & \text{if } u = s \\ -|T| & \text{if } u = 0 \\ 0 & \text{sinon} \end{cases} \quad (3.2)$$

$$\sum_{k=1}^m x_{0v}^k = 1 \quad \forall v \in T \quad (3.3)$$

$$x_{uv}^k \leq C_{uv} t_{uv}^k \quad \forall (u, v) \in A, \forall k \in K \quad (3.4)$$

$$t_{uv}^k + t_{vu}^k \leq 1 \quad \forall \{u, v\} \in E, \forall k \in K \quad (3.5)$$

$$t_{uv}^{k+1} \leq t_{uv}^k \quad \forall (u, v) \in A_1, \forall k \in K, k < m \quad (3.6)$$

$$t_{uv}^{k+1} + t_{vu}^{k+1} \leq t_{uv}^k + t_{vu}^k \quad \forall \{u, v\} \in E, \forall k \in K, k < m \quad (3.7)$$

$$\sum_{u \in P(v)} \sum_{k=1}^m z_{uvw}^{kk'} = x_{vw}^{k'} \quad \forall (v, w) \in A, v \neq 0, \forall k' \in K \quad (3.8)$$

$$\sum_{w \in S(v)} \sum_{k'=1}^m z_{uvw}^{kk'} = x_{uv}^k \quad \forall (u, v) \in A, v \neq s, \forall k \in K \quad (3.9)$$

$$z_{uvw}^{kk'} \leq \min(C_{uv}, C_{vw}) y_{uvw}^{kk'} \quad \forall (u, v), (v, w) \in A, \forall k, k' \in K \quad (3.10)$$

$$\sum_{w \in S(v)} \sum_{k'=1}^m y_{uvw}^{kk'} = t_{uv}^k \quad \forall (u, v) \in A, v \neq s, \forall k \in K \quad (3.11)$$

$$x_{uv}^k \geq 0 \quad \forall (u, v) \in A, \forall k \in K \quad (3.12)$$

$$z_{uvw}^{kk'} \geq 0 \quad \forall (u, v), (v, w) \in A, \forall k, k' \in K \quad (3.13)$$

$$t_{uv}^k \in \{0, 1\} \quad \forall (u, v) \in A, \forall k \in K \quad (3.14)$$

$$y_{uvw}^{kk'} \in \{0, 1\} \quad \forall (u, v), (v, w) \in A, \forall k, k' \in K \quad (3.15)$$

La contrainte 3.2 exprime la conservation du flot, la contrainte 3.3 assure qu'une seule unité de flot soit générée par une turbine, la contrainte 3.4 oblige le flot à respecter les capacités sur les arcs, la contrainte 3.5 rend impossible au flot de circuler dans deux sens à la fois (l'ensemble E représentant les arêtes du réseau), la contrainte 3.6 impose qu'un $(k+1)$ ième lien parallèle soit utilisé seulement si le k ième lien avant lui est utilisé aussi, les contraintes 3.8 à 3.11 concernent la non bifurcation. (Pour plus de détail voir Hertz *et al.* (2012)).

Ce modèle possède une propriété intéressante qui sera utilisée dans le prochain chapitre : il est possible de résoudre le modèle pour un sous-ensemble de turbines inclus dans T et d'ajuster les capacités C_{uv} selon l'utilisation partielle qui a été effectuée. Il est alors possible de résoudre à

nouveau le problème pour un nouveau sous-ensemble de turbines. Cette propriété permet donc de fractionner le problème et d'obtenir, par connexion de sous-ensembles de turbines successifs, une solution qui ne sera pas nécessairement optimale, mais qui sera probablement bonne tout de même. Ceci sera discuté en détail dans le chapitre 4 qui porte sur la résolution.

3.3 La formulation dans le graphe adjoint

Il convient tout d'abord de définir ce qu'est un graphe adjoint :

Le graphe adjoint associé à un graphe G est le graphe, noté $L(G)$, dont les sommets représentent les arcs de G , et tel que deux sommets de $L(G)$ (deux arcs dans G) sont reliés par un arc dans $L(G)$ si et seulement si l'extrémité du premier arc dans G est l'origine du deuxième arc dans G .

La figure 1.2 représente une solution du graphe explicité dans la figure 1.1. Cette solution contient un cycle défini par les noeuds 1,9,10,11,12,13. Hertz *et al.* (2012) ont montré qu'une solution dans G est une anti-arborescence dans $L(G)$.

En effet, la solution de la figure 1.2 dans le graphe adjoint est celle explicitée dans la figure 3.1 et représente bien une anti-arborescence. On voit bien la structure arborescente de la solution dans ce graphe. Mentionnons que cette représentation dans le graphe adjoint n'est pas unique : au noeud 12, il y a deux chemins possibles pour les turbines 5,6,7,8, par exemple. En effet, les deux unités provenant de 6 et 7 peuvent être remplacées par les deux unités provenant de 5 et 8. Normalement le vrai graphe adjoint ne contiendrait pas les noeuds turbines en triangle, ni celui de la sous-station en carré ; on les ajoute car cela permet de mieux voir d'où vient et où va le courant.

De façon formelle, l'ensemble des noeuds du graphe adjoint V_ℓ est l'union disjointe de T (l'ensemble de turbines), L (l'ensemble des liens pour chaque arc $(u, v) \in A$ de la forme (u, v, k)) et $\{s\}$ (la sous-station). L'ensemble A_ℓ est l'union disjointe des 3 catégories d'arcs suivantes :

1. tous les arcs de la forme $(u, (u, v, k))$ pour $u \in T, (u, v, k) \in L$;
2. tous les arcs de la forme $((u, v, k), (v, w, k'))$ pour $(u, v, k), (v, w, k') \in L$ et
3. tous les arcs de la forme $((u, s, k), s)$ pour $(u, s, k) \in L$.

Le graphe adjoint est donc $L(G) = (V_\ell, A_\ell)$. Un lecteur attentif dirait que ce n'est pas exactement le graphe adjoint selon la définition ci-haut ; celui-ci comporterait en effet seulement la deuxième catégorie d'arcs et ne contiendrait pas les turbines et la sous-station dans ses noeuds. Ces ajouts au véritable graphe adjoint sont cosmétiques et ne servent qu'à montrer d'où le courant provient et où il s'achemine comme sur la figure 3.1 ; le terme graphe adjoint continuera d'être celui utilisé par abus de langage dans la suite pour désigner le graphe (V_ℓ, A_ℓ) .

3.3.1 Les coûts, les capacités et les contraintes

Les coûts, les capacités et les contraintes restent les mêmes, sauf qu'ils s'appliquent désormais aux noeuds de l'ensemble L . En effet, le coût c_{uv}^k correspond au coût du noeud $(u, v, k) \in L$, la capacité C_{uv} s'applique à tous les noeuds $(u, v, i), 0 \leq i \leq m - 1$ et les contraintes sur les liens

parallèles agissent sur les noeuds de l'ensemble L . Un coût de 0 et une capacité infinie sont assignés aux noeuds de l'ensemble $T \cup \{s\}$.

3.3.2 Quelques définitions

Une façon de traiter les contraintes de façon à obtenir une formulation proche de celle de l'arbre de Steiner est de les enfermer dans la fonction de coût.

Soit la fonction $f_{\mathcal{T}} : V_{\mathcal{T}} \rightarrow \mathbb{N}$ qui retourne le nombre de turbines qui sont atteignables depuis un noeud de l'anti-arborescence \mathcal{T} en remontant celle-ci à partir du noeud. Par exemple, sur la figure 3.1 $f_{\mathcal{T}}((11, 12, 0)) = 2$, soit les deux turbines 3 et 4.

Soit la fonction de coût

$$w_{\mathcal{T}}(u, v, k) = \begin{cases} \infty & \text{si } f_{\mathcal{T}}(u, v, k) > C_{uv} \\ \infty & \text{si } \exists i, 0 \leq i < k \text{ tel que } f_{\mathcal{T}}(u, v, i) = f_{\mathcal{T}}(v, u, i) = 0 \\ \infty & \text{si } (v, u, k) \in V_{\mathcal{T}} \cap L \\ c_{uv}^k & \text{sinon} \end{cases} \quad (3.16)$$

où $(u, v, k) \in V_{\mathcal{T}} \cap L$.

La première partie de la définition fonction de coût ($f_{\mathcal{T}}(u, v, k) > C_{uv}$) interdit de dépasser la capacité, la deuxième ($f_{\mathcal{T}}(u, v, l) = f_{\mathcal{T}}(v, u, l) = 0, 0 \leq l < k$) interdit de prendre un lien parallèle avant ses prédécesseurs et la troisième partie interdit toute solution admettant du courant circulant dans les deux sens sur un même lien.

Il est possible alors de donner une définition d'une solution réalisable et optimale dans le cadre théorique du problème de l'arbre de Steiner.

Definition 1. Une solution du problème de la conception de parc éolien est un ensemble de sommets $V_{\mathcal{T}} \subseteq V_{\ell}$ et d'arcs $A_{\mathcal{T}} \subseteq A_{\ell}$ où $\mathcal{T} = (V_{\mathcal{T}}, A_{\mathcal{T}})$ est un anti-arbre dans G avec $T \subset V_{\mathcal{T}}$.

Definition 2. Une solution réalisable du problème de la conception est une solution dont le coût total est fini :

$$\sum_{(u,v,k) \in V_{\mathcal{T}}} w_{\mathcal{T}}(u, v, k) < \infty. \quad (3.17)$$

Definition 3. Une solution optimale du problème de la conception est une solution dont le coût total est minimisé. Mathématiquement, le coût total est représenté par cette expression :

$$\min_{\mathcal{T} \in G} \sum_{(u,v,k) \in V_{\mathcal{T}}} w_{\mathcal{T}}(u, v, k). \quad (3.18)$$

Il devient clair avec la définition 3 que trouver la solution optimale du problème de la conception d'un parc éolien équivaut à trouver l'arbre de Steiner \mathcal{T} du problème de Steiner lorsque les coûts des arcs sont définis par la fonction $w_{\mathcal{T}}$. Dans le problème de l'arbre de Steiner, par contre, la fonction

de coût ne dépend pas de \mathcal{T} , c'est la différence majeure entre ce problème classique et celui qui est traité ici.

La solution présentée dans les figures 1.2 et 3.1 satisfait les définitions 1 et 2, mais ne satisfait pas la définition 3, car elle est de coût 34 alors qu'il existe d'autres solutions moins coûteuses. La solution présentée dans les figures 3.2 et 3.3 satisfait la définition 3, car elle coûte 30 (2 unités de coût pour les liens noirs et 1 pour les liens gris) et il n'existe pas de solution moins coûteuse, c'est une solution optimale.

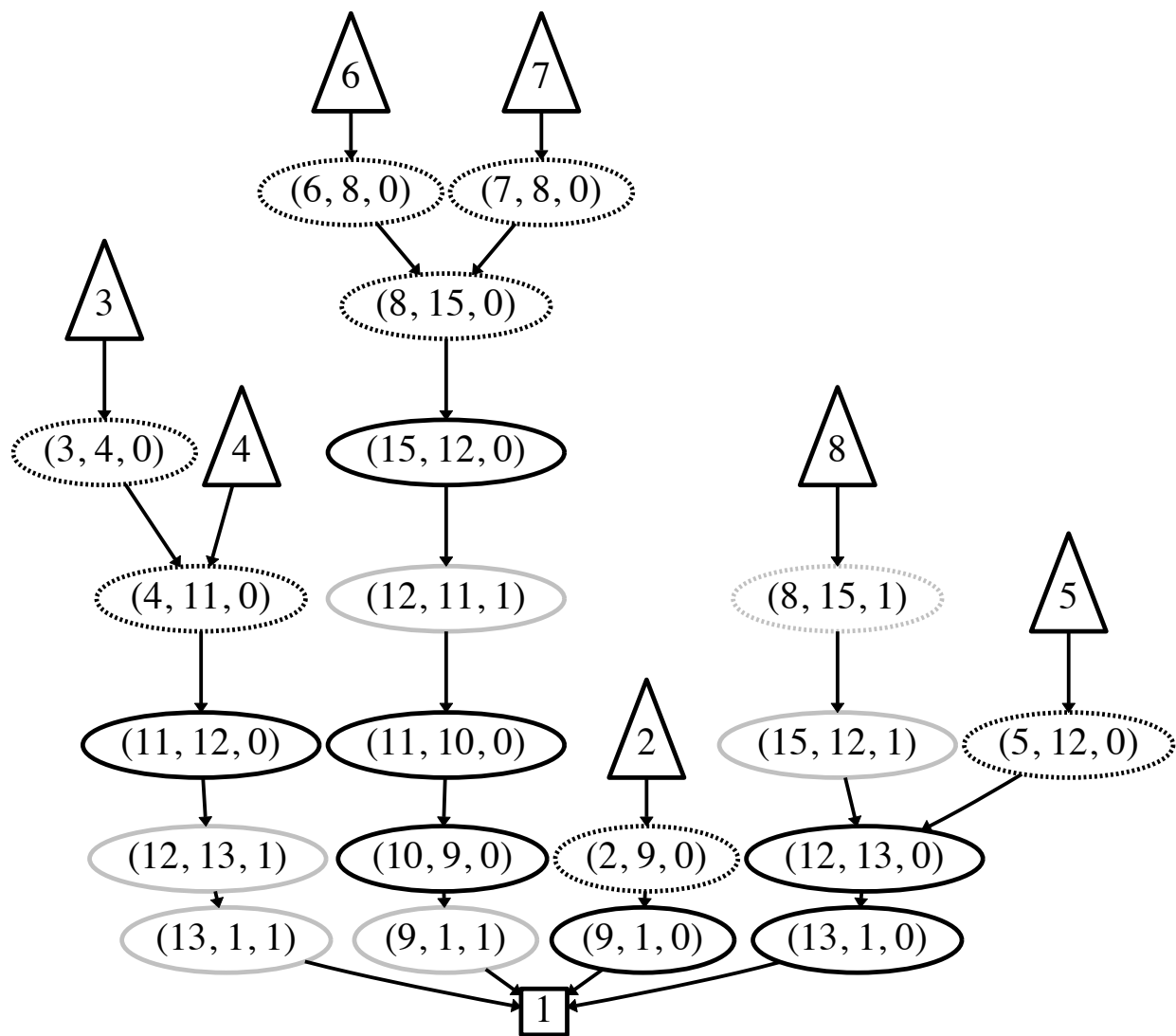


Figure 3.1 Une solution de la figure 1.2 dans le graphe adjoint.

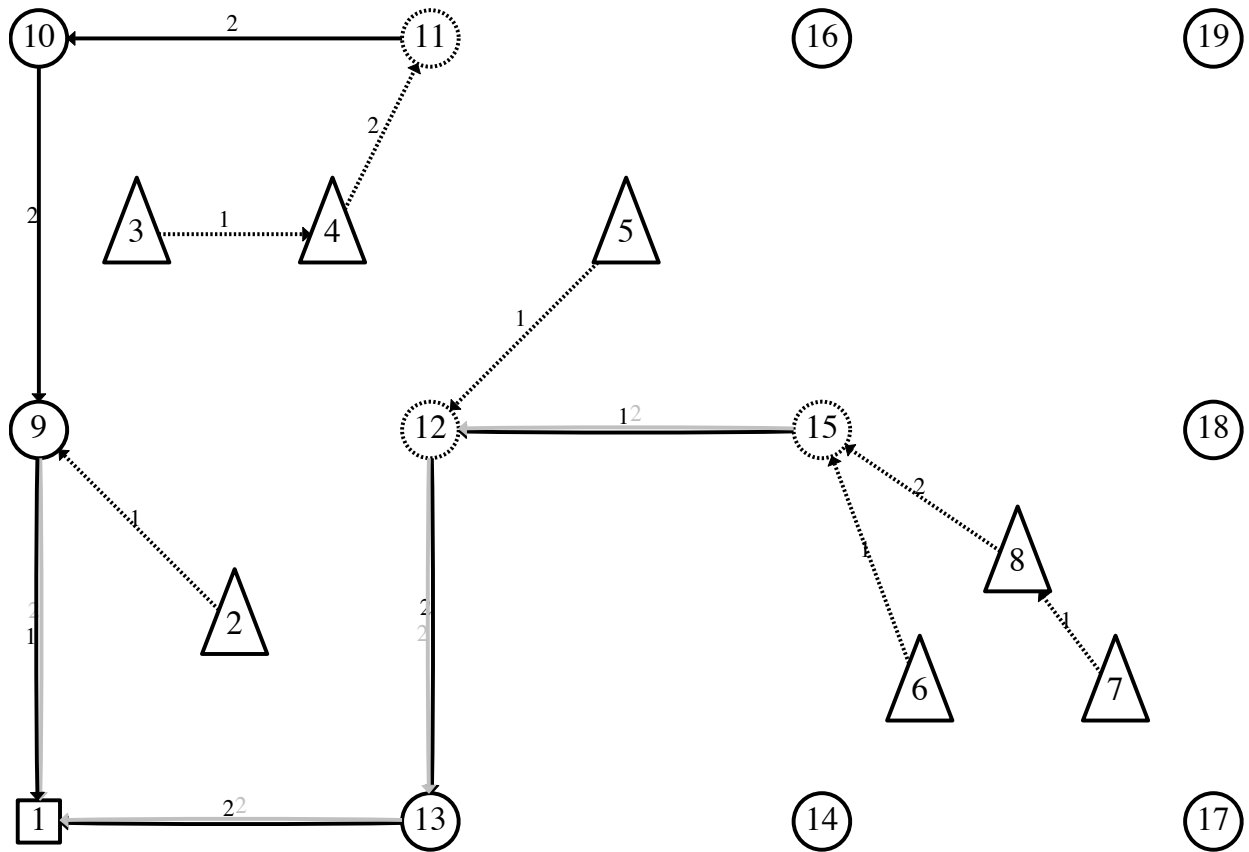


Figure 3.2 La solution optimale de coût 30 dans le graphe original.

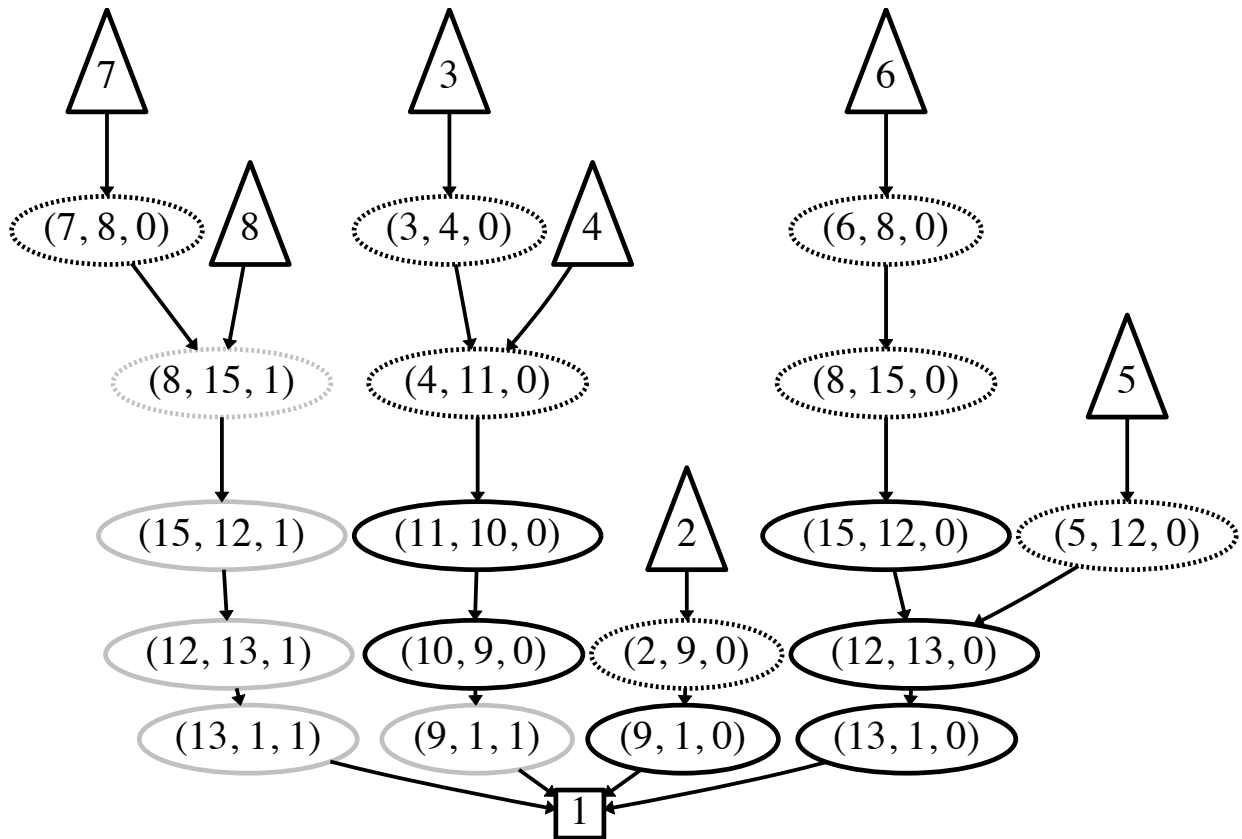


Figure 3.3 L'anti-arborescence optimale dans le graphe adjoint.

CHAPITRE 4

RÉSOLUTION

L'inspiration de la technique de résolution provient de l'article de Gendreau *et al.* (1998), quoiqu'il y a une grande différence : étant donné la complexité d'une reconnexion, au lieu d'utiliser un algorithme simple comme Kruskal à la manière de Gendreau *et al.* (1998) et lui faire des adaptations complexes pour le problème traité ici, l'algorithme exact de Hertz *et al.* (2012) sous-traitera les reconnexions et gèrera donc toute la complexité de celles-ci avec la rapidité qu'on connaît à CPLEX.

La complexité d'une reconnexion provient du fait que la solution du graphe adjoint est une anti-arborescence contenant des noeuds représentant des liens parallèles. Il est alors difficile d'appliquer naïvement la méthode de Gendreau *et al.* (1998), car le retrait d'un noeud représentant un lien parallèle sera permis seulement si ce noeud correspond au dernier lien parallèle utilisé (le noeud $(u, v, k+1)$ doit être retiré avant le noeud (u, v, k)). La stratégie également d'opérer une reconnexion grâce à l'algorithme de Kruskal ou Prim est inapplicable, car l'électricité qui parcourt les arcs a une direction anti-arborescente (vers la sous-station) et Kruskal ou Prim ne tiennent pas compte de cette contrainte.

4.0.3 La construction d'une solution initiale réalisable

En gros, Gendreau *et al.* (1998) construisent d'abord une solution initiale à partir de l'heuristique de Takahashi et Matsuyama (1980), c'est-à-dire en prenant une suite de plus courts chemins entre les noeuds obligatoires, pour ensuite l'améliorer avec un premier tabou. Ici, la construction de la solution initiale s'inspire également de l'heuristique de Takahashi et Matsuyama (1980), sauf qu'au lieu de faire les plus courts chemins successivement entre deux noeuds obligatoires, les plus courts chemins successifs se font chaque fois entre une turbine et la sous-station. Construire la solution initiale en prenant simplement un plus court chemin entre deux noeuds obligatoires pose problème : d'abord il n'est pas toujours possible de connecter une turbine à une autre avec un plus court chemin, car les relais empêchent de passer du réseau hors-terre au réseau sous-terre, et enfin, même si cela était possible, le sens du courant ne serait pas défini une fois le plus court chemin réalisé, il n'est défini que lorsqu'il y a un chemin entre une turbine et la sous-station.

De façon semblable à Takahashi et Matsuyama (1980), on utilisera l'algorithme du plus court chemin de Dijkstra (1959) pour relier les sommets obligatoires à la sous-station.

La procédure est explicitée en l'algorithme 1. Elle prend itérativement une nouvelle turbine de l'ensemble T et la connecte avec la sous-station s par le plus court chemin entre t et s selon la fonction de coût $w_{\mathcal{T}}$; elle rajoute ce chemin à l'arbre courant \mathcal{T} et recommence. Par exemple, la solution de la figure 1.2 a été générée en prenant successivement le plus court chemin de 3 à s , 4 à s ,

2 à s , 5 à s , 8 à s , 6 à s et 7 à s et en mettant à jour le coût (une fois qu'un noeud, correspondant à un lien dans le graphe original, est utilisé, il n'a plus à être payé, un coût de 0 doit lui être assigné) et la capacité pour chaque noeud.

Entrées : $L(G), T$

Sorties : \mathcal{T}

```

1   $\mathcal{T} \leftarrow \emptyset$ ;
2   $\tau \leftarrow \emptyset$ ;
3  tant que  $\tau \neq T$  faire
4      Prendre une turbine  $t$  dans l'ensemble  $T - \tau$  ;
5      Ajouter cette turbine dans l'ensemble  $\tau$ ;
6      Prendre le plus court chemin entre  $t$  et  $s$ ;
7      Entreposer ce chemin dans  $p$ ;
8      Ajouter  $p$  à  $\mathcal{T}$  ;
9      Décroître d'une unité la capacité des noeuds du chemin  $p$ ;
10     Mettre à 0 le coût constant  $c_{uv}^k$  des noeuds dans  $p$  :  $c_{uv}^k = 0, \forall (u, v, k) \in p$ ;
11     Réinitialiser  $w_{\mathcal{T}}(u, v, k)$  des noeuds en fonction des changements de capacité et de
        coût constant;
12 fin
```

Algorithme 1: Trouver une solution initiale

Le plus court chemin aura tendance à éviter les coûts infinis, mais l'algorithme 1 reste aveugle : les plus courts chemins successifs peuvent mener à un cul-de-sac pour une nouvelle turbine où tous les chemins de cette turbine à la sous-station contiendront des sommets valant ∞ . Supposons par exemple que c_{91}^0 est très grand et que les autres coûts sont inchangés dans notre exemple du début (figure 1.1). Les premiers plus courts chemins auront tendance à éviter l'arc (9, 1). Dans ce cas, l'ordre de tout à l'heure (3 à s , 4 à s , 2 à s , 5 à s , 8 à s , 7 à s et 6 à s) mènerait à un cul-de-sac, car les turbines 3, 4, 2, 5 éviteraient l'arc (9, 1) et passeraient toutes sur (13, 1) ; il ne serait alors plus possible de connecter la turbine 6 sans enfreindre les contraintes et provoquer un coût infini (voir figure 4.1). À noter : les algorithmes développés dans ce mémoire parcourent un espace de solutions réalisables ; l'idée de mettre des pénalités à des infractions au niveau des capacités par exemple n'a pas été envisagée.

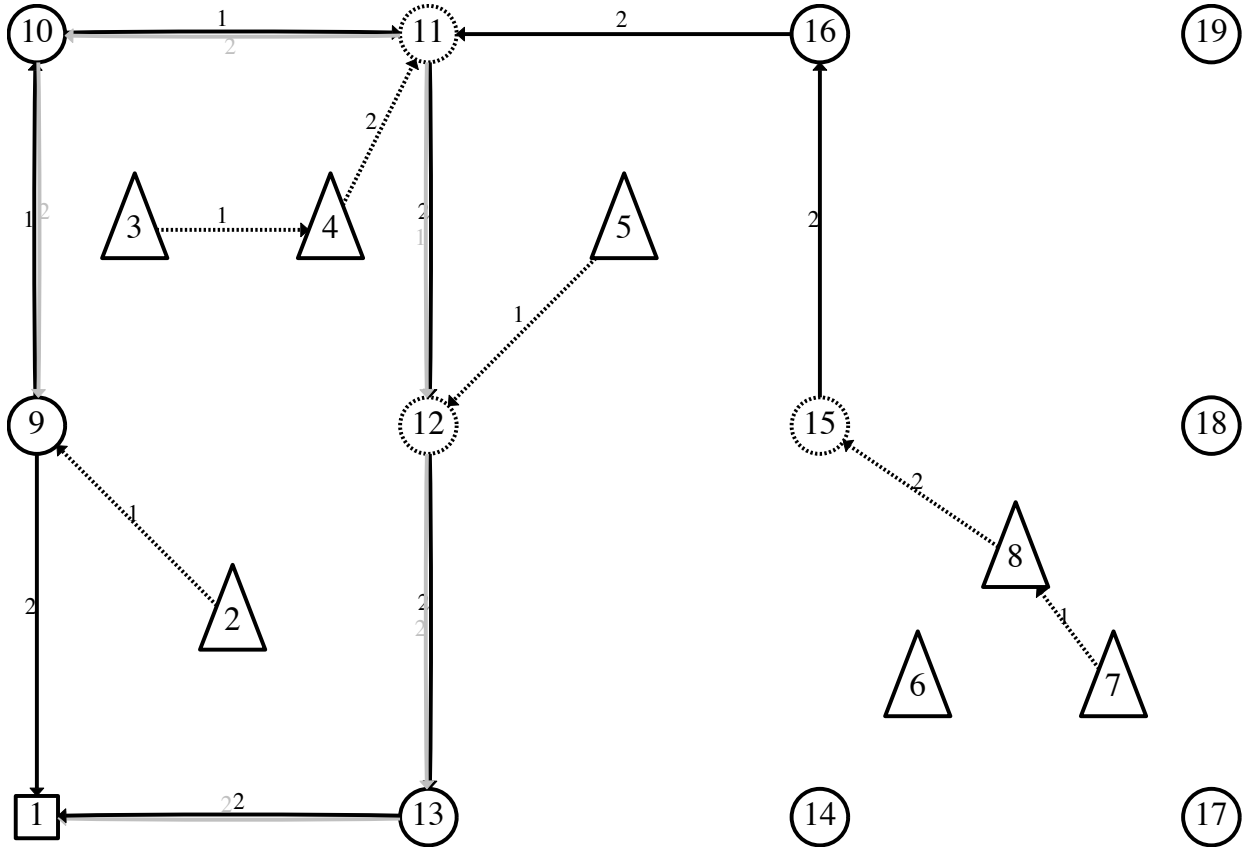


Figure 4.1 Il est clair qu'il est impossible de connecter la turbine 6 si toutes les turbines 3,4,2,5 évitent de passer par (9,1) dans l'ordre de connexion 3,4,2,5,8,7,6 dans l'algorithme 1. Par contre, si nous avons choisi l'ordre 6,7,8,5,3,4,2 pour nos plus courts chemins dans l'algorithme 1, la solution résultante serait réalisable.

La figure 4.1 suggère que l'ordre dans lequel sont prises les turbines lors de l'algorithme 1 peut contribuer à la découverte d'une solution réalisable ou pas. L'ordre 6,7,8,5,3,4,2 donne une solution réalisable, mais l'ordre 3,4,2,5,8,6,7, non, si on admet un coût élevé pour l'arc (9,1).

Pour surmonter cette difficulté, une technique d'évaluation rapide des ordres a été développée ; elle sera couplée avec une technique de réparation d'un ordre.

Une technique d'évaluation rapide

Il serait intéressant de pouvoir tester plusieurs ordres rapidement pour surmonter l'irréalisabilité de certains ordres et pour tenter de trouver des ordres proches de la solution optimale. Hélas, certaines instances du réseau $L(G)$ contiennent jusqu'à environ 10 000 arêtes, 2000 sommets et une centaine de turbines. On se rend rapidement compte que tester seulement 100 ordres (ce qui est peu : pour 100 turbines, il y a en effet 100! ordres différents) implique d'appliquer 10 000 fois l'algorithme

de plus court chemin dans un réseau comportant 10 000 arêtes.

Il est possible de suggérer une version accélérée de l'algorithme 1. Ceci se fait grâce à deux améliorations :

Retirer l'information inutile

Sur les instances traitées ici, les noeuds de l'ensemble L ont un degré généralement supérieur à 16. De plus, lorsqu'un sommet (u, v, k) à une itération de l'algorithme 1 satisfait

$$f_{\mathcal{T}}(u, v, k) = C_{uv} \quad \text{ou} \quad (4.1)$$

$$(v, u, k) \in V_{\mathcal{T}} \cap L, \quad (4.2)$$

alors, $w_{\mathcal{T}}(u, v, k) = \infty$, et ce coût ne changera plus lors des itérations suivantes de l'algorithme 1.

Rien ne sert alors de garder ce noeud dans le graphe $L(G)$ lors des prochaines itérations de l'algorithme. La suppression de ce noeud supprimera donc également au moins 16 arêtes de $L(G)$, ce qui réduira la complexité de l'algorithme de plus court chemin utilisé à chaque itération suivante. Ces noeuds peuvent être supprimés de façon permanente.

Étant donné l'omniprésence de ces noeuds, les gains d'efficacité computationnelle sont substantiels. Cependant, leur omniprésence menace parfois aussi la quête d'une solution initiale réalisable.

Condenser l'information

Il serait intéressant de pouvoir réduire le nombre de turbines dans l'ensemble T . Si le nombre total de turbines à considérer passe de 100 à 50 par exemple, le nombre d'ordres différents à tester passe de $100!$ à $50!$, une amélioration considérable. Il n'est pas irréaliste de considérer cette voie, car les turbines sont souvent concentrées en groupes géographiques, par exemple, le groupe $\{3, 4\}$ et le groupe $\{6, 7, 8\}$ dans le graphe de la figure 1.1. Il suffit de considérer une seule superturbine agrégée représentante de toutes les autres d'un groupe géographique pour réduire le nombre de turbines à traiter. Le nombre de turbines agrégées dans une superturbine ne doit pas dépasser la capacité maximale dans le graphe, car sinon il pourrait arriver que le plus court chemin de la superturbine passe par des liens de capacités insuffisantes. Par exemple, les turbines 6,7,8 dans 1.1 ne peuvent être toutes agrégées, car la capacité des arcs dans le réseau est 2 et non 3. Si elles étaient agrégées, un courant de 3 unités sortirait de la superturbine et dépasserait la capacité de l'arc $(8, 15)$ ou $(6, 15)$ par lequel il est obligé de passer.

Pour éviter d'agréger trop de turbines, on résout avec CPLEX le problème pour chaque groupe de turbines grâce au modèle 3.2 de Hertz *et al.* (2012). Aucun groupe de turbines n'a nécessité plus de 1 seconde de calcul. La figure 4.2 montre la résolution du modèle 3.2 pour le groupe 6,7,8 par CPLEX. La solution que fournit CPLEX met 7 et 8 ensemble (cela aurait pu être 6 et 8 ou 6 et 7 qui sont deux autres solutions optimales), on va donc créer une superturbine $\{7, 8\}$ et laisser la turbine 6 comme elle est.

Plus formellement, l'agrégation des groupes se fait comme suit : sans égard à la capacité pour

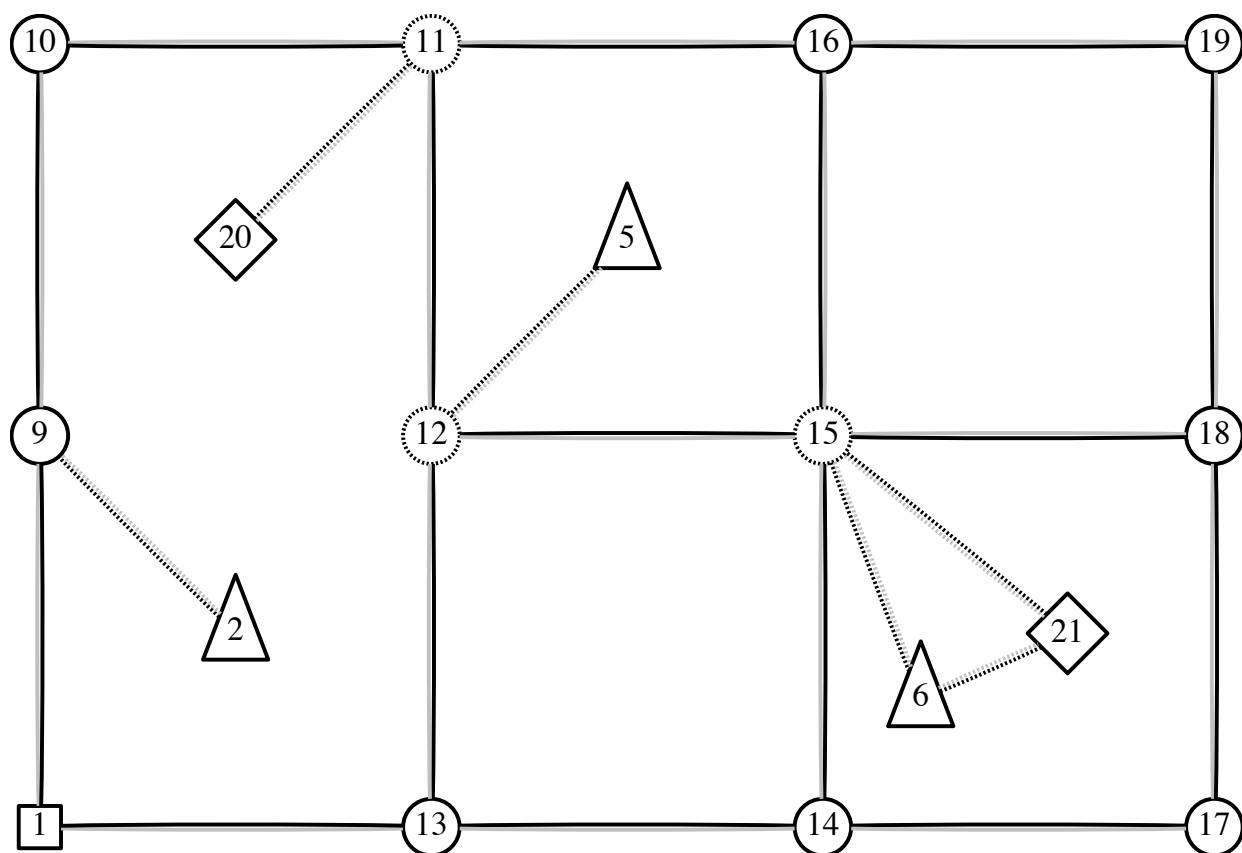


Figure 4.3 Le graphe de la figure 1.1 après agrégation.

Reste qu'il se peut que l'agrégation établie ne soit pas celle de la solution optimale. Pour retrouver une partie de l'espace solution perdu, il faudra faire une post-optimisation qui réoptimisera les turbines lors de leur désagrégation. Avant de discuter de cette réoptimisation, il faut élaborer une technique de réparation d'un ordre.

Une technique de réparation d'un ordre

Certains ordres mènent vers des *culs-de-sac*, c'est-à-dire qu'à partir de la k -ième turbine pour un ordre de n turbines par exemple (où $k \leq n$), l'algorithme 1 n'a plus de chemin vers la sous-station ne contenant pas de lien de coût ∞ .

Une technique naïve de réparation a été prévue. Lorsqu'il n'y a plus de chemin de coût fini d'une turbine vers la sous-station, un plus court chemin entre la turbine et la sous-station est calculé en supposant que $\infty + \infty > \infty$, c'est-à-dire que le plus court chemin doit passer par le moins d'arêtes interdites possible. À partir du premier lien interdit rencontré, il faut retracer toutes les turbines qui passent par ce lien et les déconnecter de l'anti-arborescence. La turbine courante est reconnectée et les turbines déconnectées sont ajoutées aux turbines à connecter à la fin de la liste.

Il y a une subtilité à gérer lors de l'utilisation d'une réparation. Elle exige que tous les liens sur un arc de la forme (u, v) ayant un lien $(u, v, k + 1)$ tel que (u, v, k) est inutilisé seront renommés en décrémentant de une unité leur numéro de lien parallèle. Il est possible d'illustrer ce cas de figure en prenant un segment de solution.

Par exemple, le segment de solution en figure 4.4 explicite une consommation de 1 unité et de 2 unités sur les liens des arcs $(1, 2)$ et $(2, 3)$. (On met des pointillés pour indiquer que le flot arrive de l'extérieur et y repart). Si on se concentre sur les noeuds 1, 2 et 3, il est possible que le segment de solution correspondant soit le segment d'arbre de la figure 4.5, où le flot sur $(1, 2, 0)$ va vers $(2, 3, 1)$ et celui sur $(1, 2, 1)$ vers $(2, 3, 0)$. Dans ce cas, un plus court chemin se butant sur le lien $(2, 3, 1)$, rendu à sa capacité maximale, enlèvera tous les noeuds issu de ce noeud, ici, les noeuds $(2, 3, 1)$ et $(1, 2, 0)$ menant à la figure 4.6. Il faudra alors décrémenter le numéro du lien parallèle $(1, 2, 1)$ à $(1, 2, 0)$ pour éviter que la réparation mène à cette solution irréalisable qui utilise un lien gris avant un lien noir.

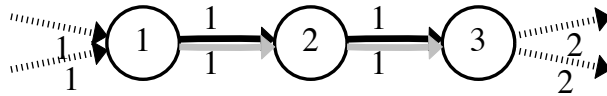


Figure 4.4

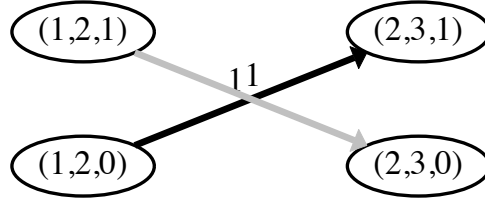


Figure 4.5

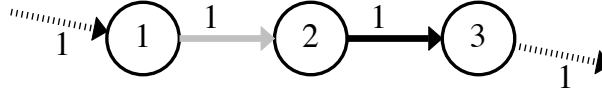


Figure 4.6

L'algorithme de construction intégrant accélération et réparation

L'algorithme 2 regroupe toutes les procédures précédemment expliquées.

4.0.4 Amélioration de la solution initiale

Le papier de Gendreau *et al.* (1998) utilise un tabou consistant essentiellement à retirer des noeuds de la solution initiale, la déconnectant, pour ensuite la reconnecter avec de nouveaux noeuds sélectionnés de façon astucieuse. Comme mentionné plus haut, cette procédure est très difficilement adaptable au cas présent en raison du sens des liens et des priorités dans le retrait de liens parallèles.

Une technique prototypique a été développée. Elle nécessitera des ajustements, si une meilleure performance est désirée.

La technique utilise l'idée de Gendreau *et al.* (1998) de déconnecter et reconnecter des noeuds. Ici, les turbines et les liens intermédiaires impliqués dans leurs connexions à la sous-station seront les seuls noeuds déconnectés et reconnectés dans le processus. Lorsqu'un sous-ensemble de turbines ainsi que leurs liens impliqués sont déconnectés d'une solution réalisable, on se retrouve avec un graphe dont les constantes c_{uv}^k et C_{uv} originales sont modifiées par la solution partielle encore connectée. Il est alors intéressant de tenter de mieux reconnecter le sous-ensemble de turbines déconnectées en résolvant le problème dans ce graphe modifié.

Le modèle exact de Hertz *et al.* (2012) sera utilisé pour cette reconnexion améliorée ; plus précisément, il fournira la reconnexion optimale. Ce modèle utilisant CPLEX est très performant pour connecter un petit nombre de turbines à la fois (voir Hertz *et al.* (2012)) ce qui est parfait pour le genre de déconnexions qui sont envisagées. Pour opérer la reconnexion avec ce modèle, il suffit

Entrées : $L(G), T$

Sorties : \mathcal{T}

```

1 Identifier chaque groupe de turbines et procéder à leur agrégation;
2 Appeler le nouvel ensemble de turbines ainsi formé  $T'$ ;
3  $\mathcal{T} \leftarrow \emptyset$ ;
4  $\tau \leftarrow \emptyset$ ;
5 tant que  $\tau \neq T'$  faire
6   Prendre une turbine  $t$  dans l'ensemble  $T' - \tau$  ;
7   Ajouter cette turbine dans l'ensemble  $\tau$ ;
8   Prendre le plus court chemin entre  $t$  et  $s$ ;
9   Entreposer ce chemin dans  $p$ ;
10  si  $p$  a un coût infini alors
11    Trouver le premier lien de coût infini en partant de  $t$  dans  $p$ ;
12    Trouver toutes les turbines  $\tau'$  passant par ce lien;
13    Déconnecter ces turbines;
14    Retirer  $p$  de  $\mathcal{T}$  Refaire le plus court chemin  $p$ ;
15     $\tau \leftarrow \tau + \tau'$ ;
16  fin
17  Décrémenter les numéros parallèles des liens de  $p$  lorsque possible;
18  Ajouter  $p$  à  $\mathcal{T}$  ;
19  Décroître d'une unité la capacité des noeuds du chemin  $p$ ;
20  Mettre à 0 le coût constant  $c_{uv}^k$  des noeuds dans  $p$  :  $c_{uv}^k = 0, \forall (u, v, k) \in p$ ;
21  Réinitialiser  $w_{\mathcal{T}}(u, v, k)$  des noeuds en fonction des changements de capacité et de
    coût constant;
22  Éliminer du graphe tous les noeuds ayant désormais un coût infini qui satisfont les
    équations (4.1) et (4.2);
23 fin

```

Algorithme 2: Construction intégrée de la solution initiale

1. de fixer des variables $y_{uvw}^{kk'}$ pour lui indiquer par où la solution partielle encore connectée passe et ainsi d'empêcher une bifurcation dans la reconnexion,
2. de modifier les capacités C_{uv} selon l'utilisation de la solution partielle encore connectée et
3. de modifier les coûts c_{uv}^k des liens en mettant à 0 ceux qui sont utilisés par la solution partielle.

L'algorithme d'amélioration de la solution initiale est présenté sous l'algorithme 3. Les paramètres α, β indiquent le nombre d'itérations et le nombre de déconnexions possibles, respectivement. On les fixe différemment selon le problème. Plus α est petit, plus β peut être grand, car les déconnexions se font plus rapidement. L'idée est d'initialiser α avec une grande valeur sans qu'il y ait trop d'impact sur le temps de calcul pour pouvoir faire beaucoup d'itération (β large). Empiriquement, fixer $\alpha \in [0.4 \cdot |T|, 0.8 \cdot |T|]$ et $\beta \in [100, 200]$ produit de bons résultats.

Entrées : $L(G), T, \alpha, \beta$

Sorties : \mathcal{T}

```

1 Générer la solution initiale  $\mathcal{T}$  par l'algorithme 2;
2 tant que  $\alpha > 0$  faire
3   Prendre un nombre aléatoire  $b$  entre 0 et  $\beta$ ;
4   Choisir aléatoirement  $b$  turbines parmi  $|T|$ ;
5   Déconnecter ces turbines de  $\mathcal{T}$ ;
6   pour chaque lien  $(u, v, k)$  qui était utilisé par les  $b$  turbines faire
7     Mettre à jour la capacité de ces liens selon l'utilisation qui en est fait sans les  $b$ 
       turbines;
8     Mettre à jour les coûts  $w_{\mathcal{T}}$  et  $c_{uv}^k$  selon le nouveau  $f_{\mathcal{T}}$ ;
9   fin
10  Reconnecter ces turbines grâce à l'algorithme de Hertz et al. (2012) en n'oubliant
    pas de lui fixer les variable  $y_{uvw}^{kk'}$  selon l'utilisation décrit par la solution partielle  $\mathcal{T}$ 
    et de lui modifier les capacités et les coûts  $c_{uv}^k$ ;
11   $\alpha \leftarrow \alpha - 1$ ;
12 fin
13 Désagréger les groupes;
14 tant que La solution s'améliore faire
15   pour chaque groupe faire
16     Déconnecter les turbines du groupe;
17     Reconnecter ces turbines grâce à l'algorithme de Hertz et al. (2012) de la même
       façon que ci-haut;
18   fin
19 fin

```

Algorithme 3: Amélioration de la solution initiale

L'algorithme 3 est en vérité une puissante recherche locale : étant donné l'utilisation d'une résolution exacte, la pire reconnexion qui puisse se faire est celle revenant au point de départ avant la déconnexion, sinon, la solution est toujours améliorée. La portée de la recherche locale n'a pas

de limite : une déconnexion qui irait jusqu'à inclure toutes les turbines reviendrait à résoudre le problème original exactement.

L'algorithme est expliqué en détail sur l'exemple utilisé tout au long du mémoire dans les figures 4.7, 4.8, 4.9, 4.10, 4.11 et 4.12. Ici, $\alpha = 1$ simplement.

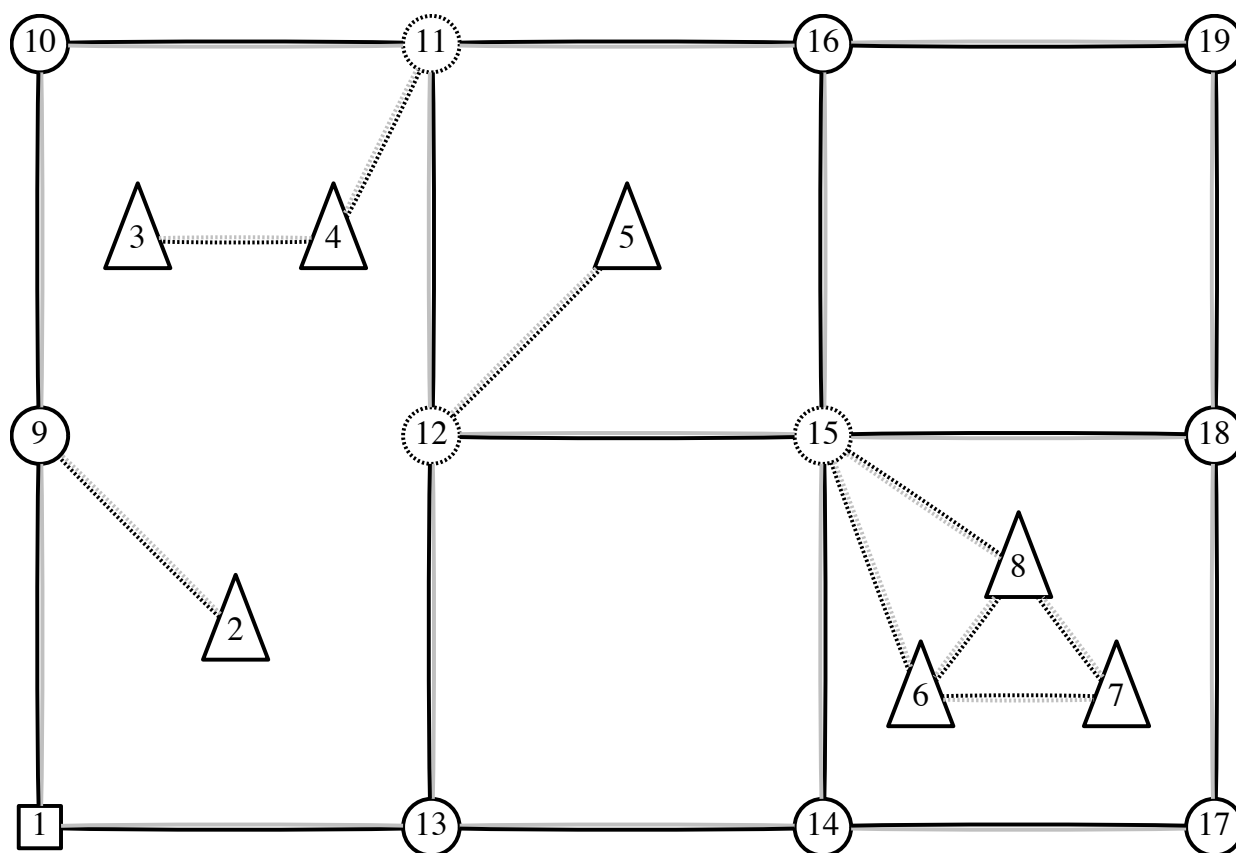


Figure 4.7 La première étape consiste à recevoir le graphe et son ensemble de turbines.

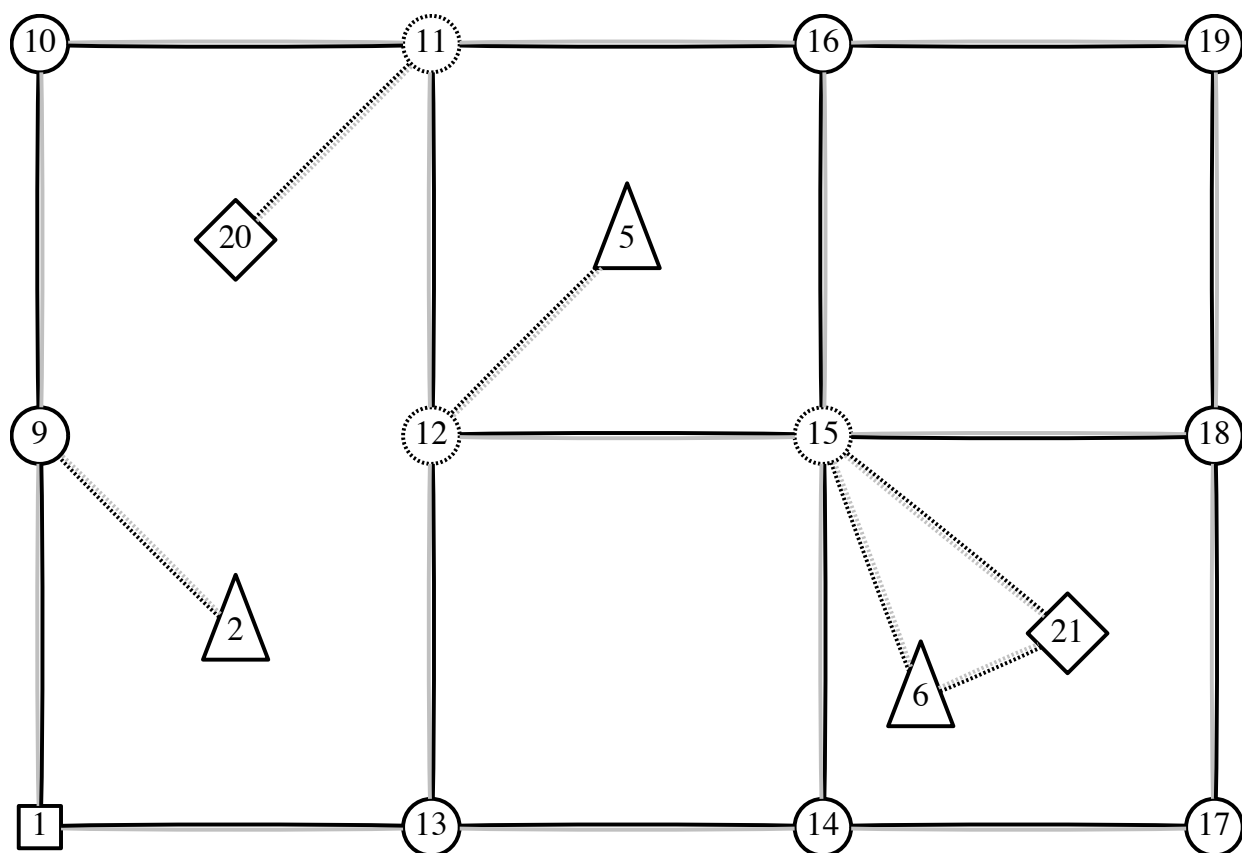


Figure 4.8 Ensuite, il faut agréger les groupes.

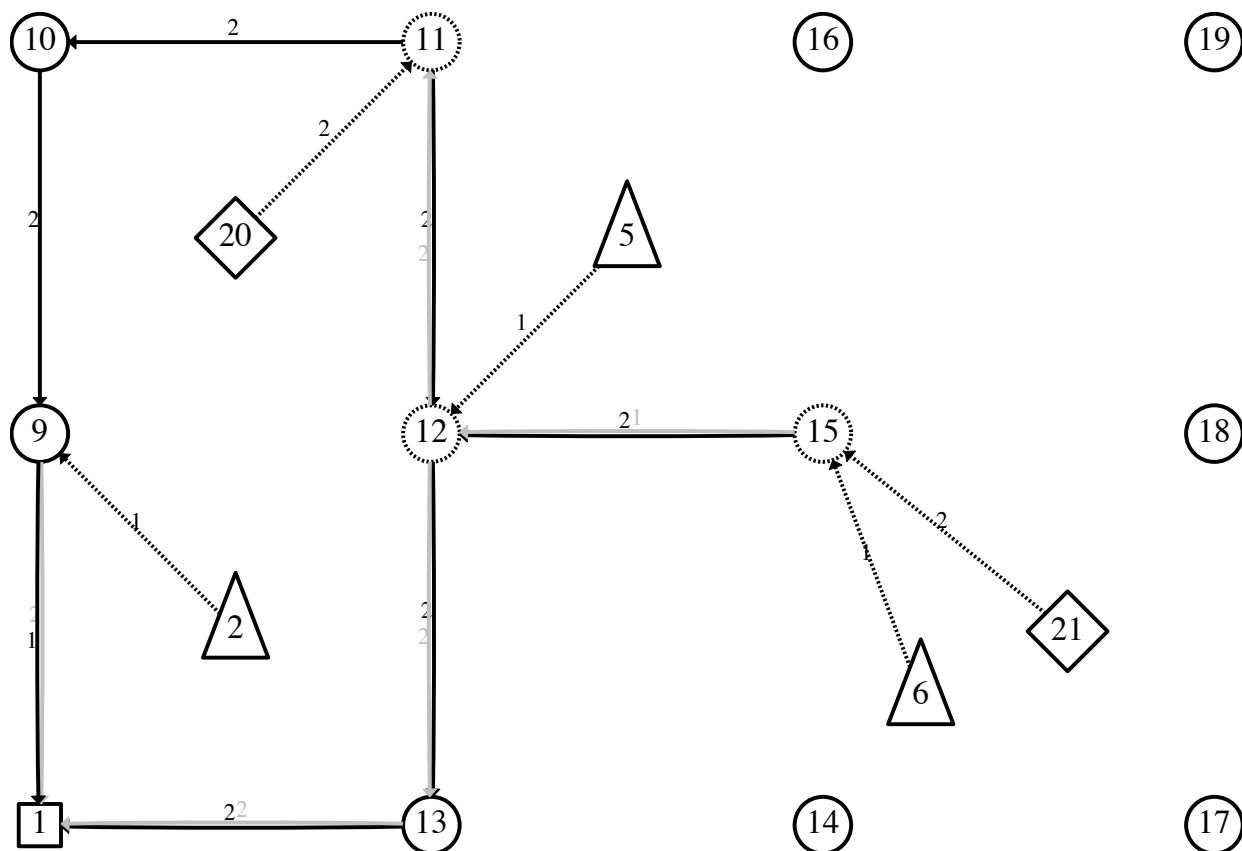


Figure 4.9 Puis, il faut générer la solution initiale grâce à l'algorithme 2. L'ordre dans lequel les turbines ont été connectées avec l'algorithme de plus court chemin est comme suit : 20,5,2,21,6. C'est une solution réalisable, il est inutile de tenter d'autres ordres.

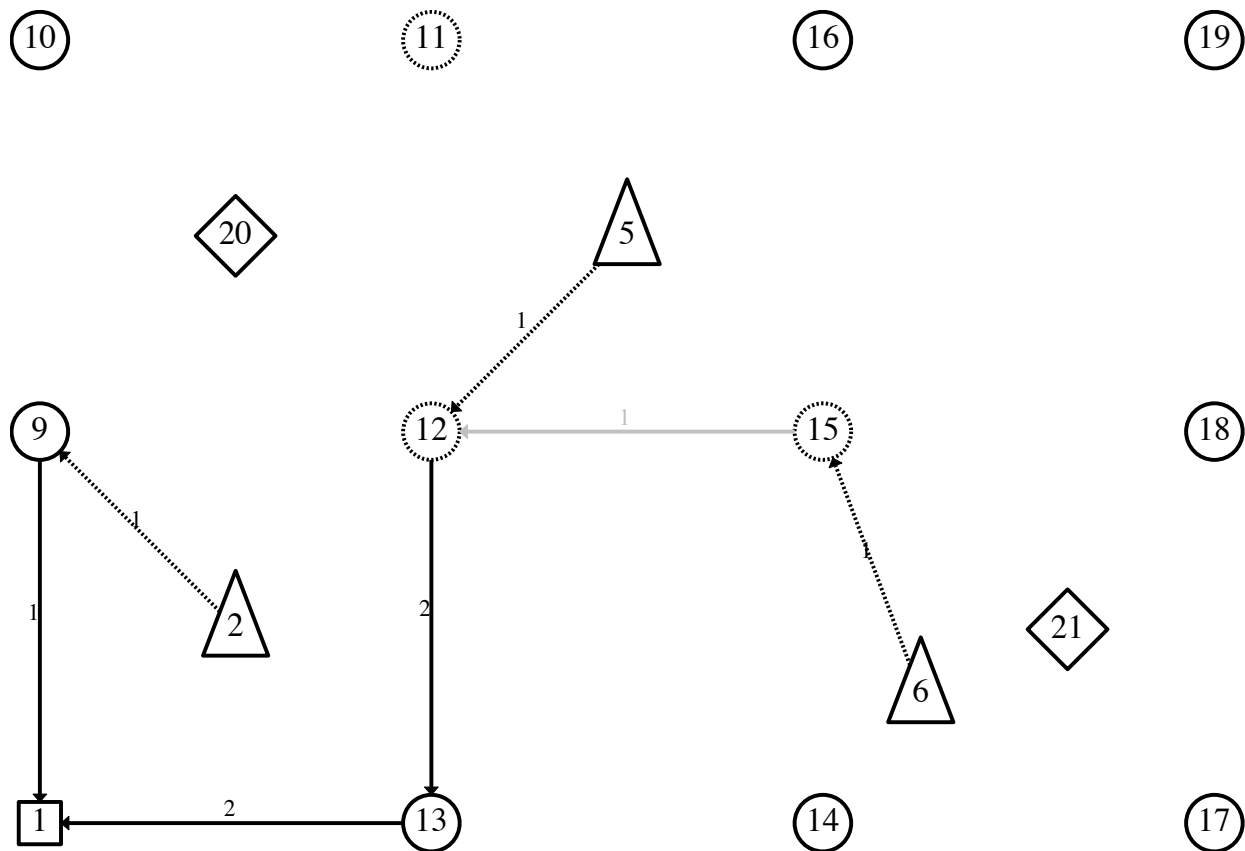


Figure 4.10 Vu qu'ici $\alpha = 1$, une seule itération de l'algorithme est opérée. Il est supposé ici que $\beta = 2$ et que les 2 turbines à déconnecter sont 20 et 21. Le lien $(15,12,1)$ est désormais interdit ; il faut le traiter comme un lien $(15,12,0)$ en décrémentant son numéro parallèle.

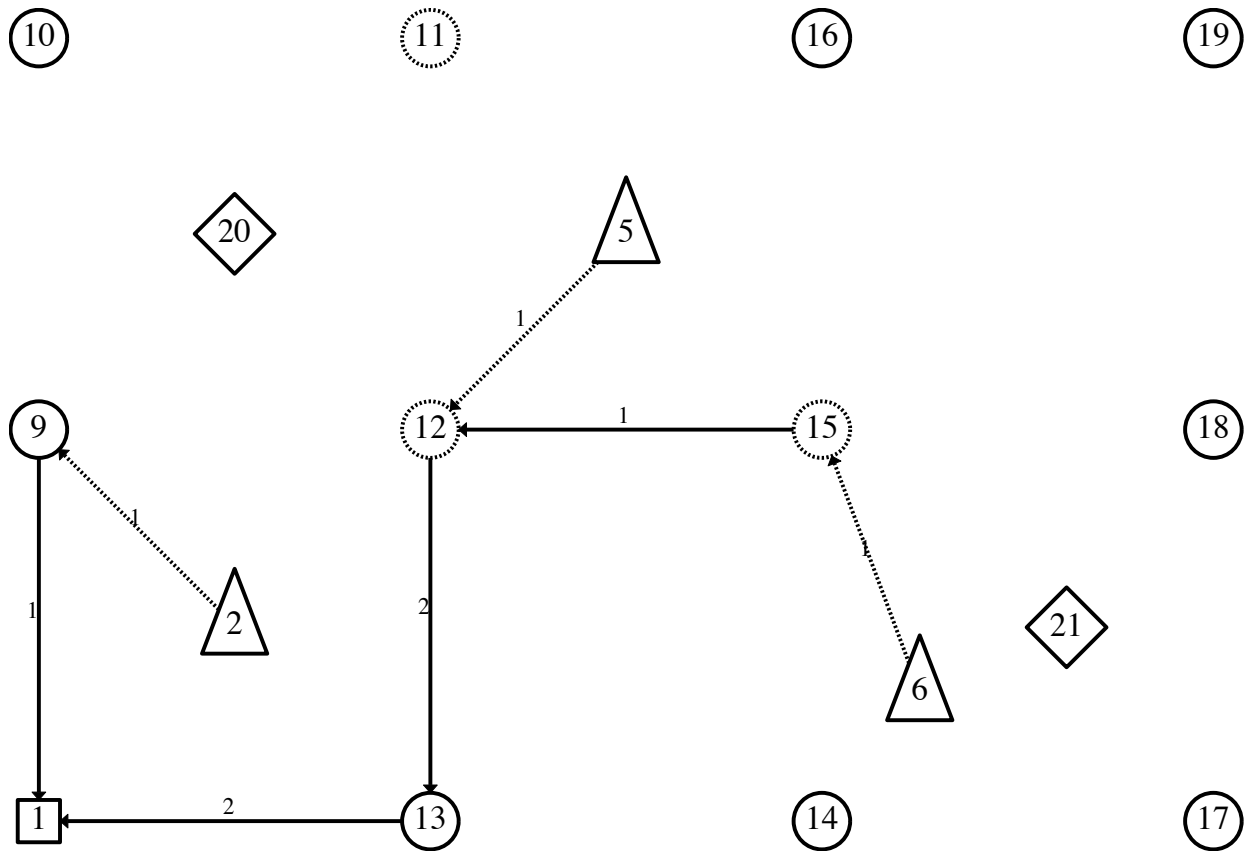


Figure 4.11 Le lien $(15,12,1)$ a été remplacé par le lien $(15,12,0)$.

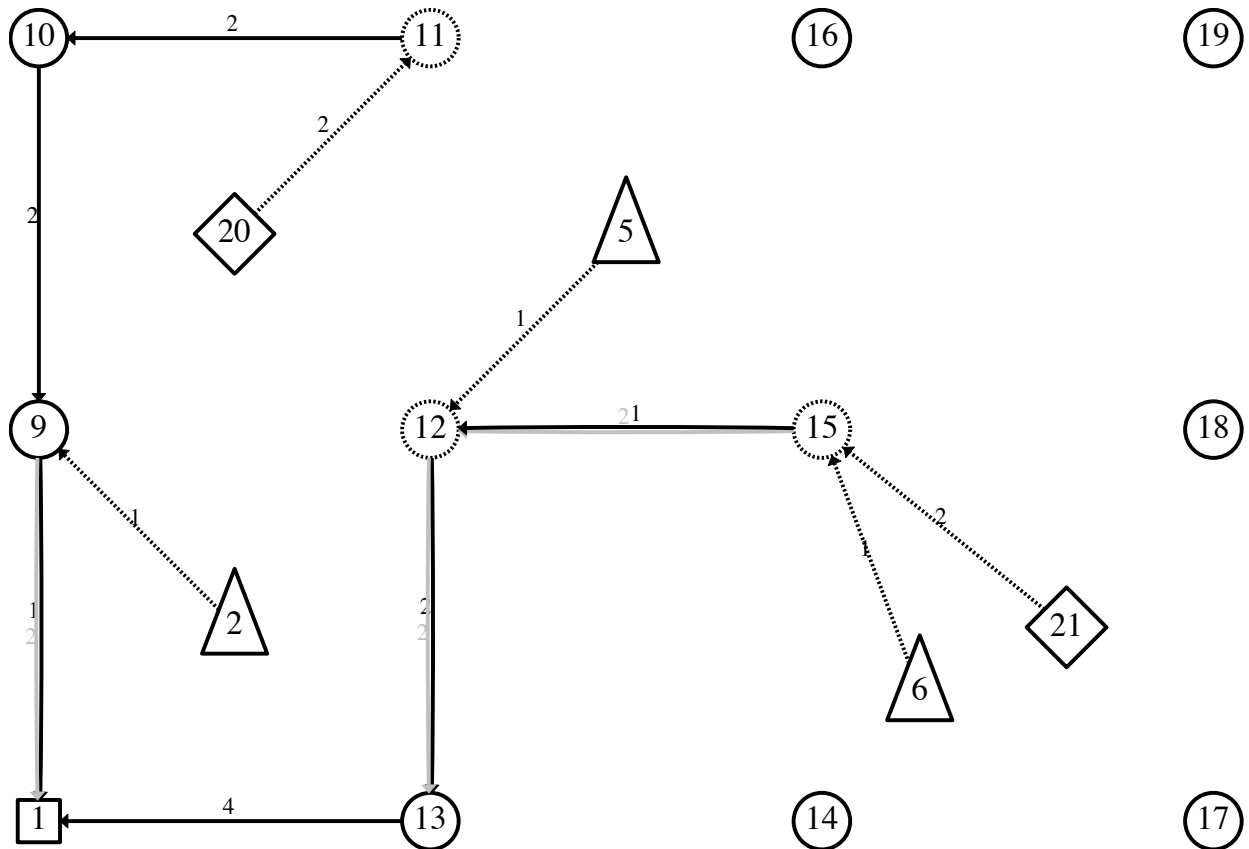


Figure 4.12 Les turbines 20 et 21 sont reconnectées de façon optimale grâce au modèle de Hertz *et al.* (2012).

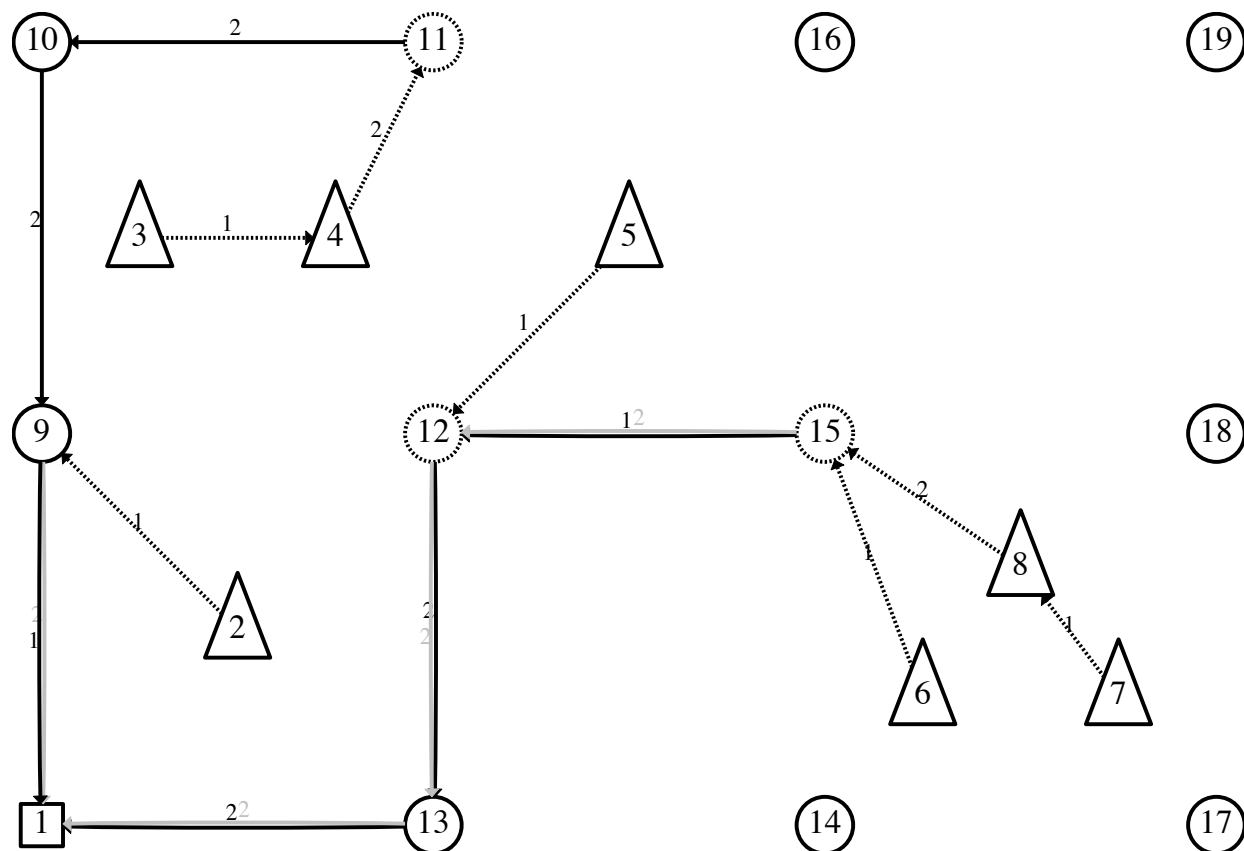


Figure 4.13 Le graphe est désagrégé. Chaque groupe de turbines est déconnecté et reconnecté par l'algorithme de Hertz *et al.* (2012) jusqu'à ce que la solution ne s'améliore plus (ici, cette procédure est inutile, car la connexions construite *a priori* des groupes était la meilleure pour la solution finale).

CHAPITRE 5

RÉSULTATS

Les résultats ont été obtenus grâce à un processeur Intel i5 de 2.5GHz avec 4GB de mémoire vive. Les procédures n'utilisant pas CPLEX ont été codées en Python.

Avant de présenter les résultats de la résolution complète par l'algorithme 3, il convient de présenter quelques chiffres (table 5.3) montrant l'utilité du processus d'agrégation utilisé dans l'algorithme 2 comparé à l'algorithme 1. Cette table met en évidence la grande multiplicité des ordres de turbines qui sont pris dans l'algorithme de construction de la solution initiale par plus courts chemins successifs. Dans le graphe original, il arrive souvent que permuter deux turbines géographiquement proche l'une de l'autre (comme 7 et 8 en figure 4.7, par exemple) ne change rien à la solution trouvée avec l'ordre précédent. L'agrégation réduit cette multiplicité ou dégénérescence et permet de visiter bien plus d'ordres réellement différents, ce qui augmente les chances de trouver des solutions réalisables, comme explicité pour les exemplaires 40,60 et 112 turbines dans la table 5.3. De plus, cette technique, en explorant davantage l'espace de solutions, permet de trouver également de meilleures solutions, comme l'exemplaire 53 turbines.

Tableau 5.1 Gains de performance résultant de l'agrégation des groupes de turbines et du processus de réparation

Turbines	Algo. 1	CPU(s)	Turbines agrégées	Algo. 3 avec 2	CPU(s)
40	?	?	27	100470.0	15
88	?	?	79	?	3600
53	182198.3	3	46	156273.8	45
73	?	?	47	?	3600
60	?	?	51	181278.4	7
112	?	?	48	214803.8	7

Les résultats sont présentés dans les tableaux 5.2, 5.3 et 5.4. Le tableau 5.2 présente le modèle de Hertz *et al.* (2012) qui a été utilisé et ce même modèle avec des coupes supplémentaires introduites dans Hertz *et al.* (2012). La raison pour laquelle le modèle de Hertz *et al.* (2012)+coupe n'est pas utilisé ici est que même si les coupes accélèrent la résolution, elles prennent du temps à générer. Étant donné que le but est de faire un nombre considérable de résolutions, elles n'ont pas été considérées. La colonne *Turbines* indique le nombre de turbines à connecter dans l'exemplaire, les colonnes n_L, m_L le nombre de noeuds et d'arêtes respectivement dans le réseau $L(G)$, les colonnes n_G, m_G la même chose dans le réseau G , ensuite les performances du modèle de Hertz *et al.* (2012) sont données pour la résolution du problème entier, puis la performance de ce même modèle avec les

coupes introduites. Dans les tableaux 5.3 et 5.4 on présente les résultats sur les mêmes exemplaires générés par l'algorithme 3 avec ou sans temps limite imposé à une reconnexion. Les paramètres α et β y sont explicités, ainsi que l'écart par rapport à la meilleure solution trouvée par les modèles exacts Hertz *et al.* (2012) ou Hertz *et al.* (2012)+coupe. Le point d'interrogation signifie que la résolution n'a pas pu trouver de solution réalisable.

Tableau 5.2 Données des exemplaires et résultat de Hertz *et al.* (2012)

Turbines	$L(G)$		G		HERTZ	CPU(<i>s</i>)	HERTZ+coupes	CPU(<i>s</i>)
	n_L	m_L	n_G	m_G				
40	143	384	689	2635	88424.8	3600	87592.4	429
88	220	517	1461	9872	126321.0	3600	126321.0	48
53	306	868	1714	8125	123941.0	3600	121779.0	2588
73	256	679	1279	5410	119887.0	3600	119887.0	598
60	189	478	1339	7778	?	?	107258.8	36086
112	281	793	1537	6127	190160.0	3600	152072.0	43700

Tableau 5.3 Résultats

Turbines	Algo. 3 avec Hertz <i>et al.</i> (2012)	α	β	CPU(<i>s</i>)	Écart(%)
40	89842.1	100	12	1356	2,6
88	?			?	
53	133419.0	40	20	1790	9,6
73	?			?	
60	119881.2	200	25	11304	11,7
112	163215.3	200	20	433	7,3

5.1 Analyse des résultats

Le problème de l'irréalisabilité de plusieurs exemplaires

La table 5.3 montre bien que l'idée naïve d'une itération de plus court chemin (algorithme 1) ne réussit pas à trouver de solutions réalisables. Même cette idée corrigée avec des agrégations et des réparations (algorithme 2) ne réussit pas toujours à atteindre la réalisabilité. Ceci indique que cette approche pour trouver une solution initiale est à revoir.

Ces algorithmes itératifs arrivent souvent, en effet, à un cul-de-sac. Pourtant, les problèmes sont loin d'être les plus difficiles pour Hertz *et al.* (2012), et l'algorithme 3 avec une reconnexion exacte aurait donc de bonnes chances de donner de bons résultats, du moins s'il pouvait avoir une solution initiale en entrée. Cet échec de l'algorithme 1 est causé par la façon itérative de construire l'arbre

Tableau 5.4 Résultats avec temps limite = 30 secondes

Turbines	Algo. 3 avec Hertz <i>et al.</i> (2012)	α	β	CPU(s)	Écart(%)
40	88549.7	100	25	197	1,1
88	?			?	
53	133500.6	40	20	539	9,6
73	?			?	
60	121880.1	200	25	4784	13,6
112	162768.0	200	20	662	7,0

solution. En effet, lorsqu'un chemin est créé entre une turbine et la sous-station, il reste dans l'arbre jusqu'à ce qu'il soit ciblé par la procédure de réparation, mais à ce moment, il peut être trop tard. C'est-à-dire que le choix de ce chemin, bien qu'attirant, peut avoir pour conséquence d'éliminer plusieurs chemins qui s'avèreraient beaucoup mieux pour supporter un ensemble de turbines et non seulement celle courante. Et étant donné que l'arbre, contraint d'être construit avec ce chemin fallacieux, contiendra une foule d'autres chemins se définissant par rapport à ce dernier, la procédure de réparation sera incapable de bien les extirper.

Il semble qu'utiliser l'algorithme de Dijkstra (1959) incite les turbines à cette sorte d'égoïsme lors de leur connexion et qu'une fois utilisé et payé, ce chemin devient faussement attrayant pour la connexion d'autres turbines qui rempliront la capacité maximale des liens et réserveront un sort funeste à la construction de la solution.

Un algorithme plus performant pourrait connecter des groupes de turbines itérativement grâce à l'algorithme de Hertz *et al.* (2012) pour réduire cet égoïsme et lors de sa procédure de réparation, il devrait non seulement déconnecter le groupe problématique, mais aussi d'autres groupes pris aléatoirement, dans l'espoir de briser la structure qui mène vers un cul-de-sac.

Les exemplaires 40 et 53

Ce sont de petites exemplaires, sans structure problématique, il est donc normal que la performance de l'algorithme 3 ne compétitionne pas celle du modèle de Hertz *et al.* (2012) qui reste efficace pour de petites exemplaires. Une meilleure solution (à 1% de l'optimum) est obtenue en imposant un temps limite à la résolution.

L'exemple 60 turbines

L'exemple 7 représente un défi pour le modèle de Hertz *et al.* (2012), c'est là que l'heuristique décrite par l'algorithme 3 devient intéressante. Alors que le modèle de Hertz *et al.* (2012) utilisé par 3 n'arrive pas à trouver une solution réalisable après une heure (3600 secondes), l'algorithme 3 trouve une solution à 20% de l'optimum (128932.1) après environ 8 minutes (494 secondes). En

laissant tourner l'algorithme plus longtemps, on obtient après environ 3 heures (11304 secondes) une solution à 11,7% de l'optimum (119881.2).

Il a été remarqué au cours de ces résolutions deux choses de premier intérêt pour l'amélioration du prototype qu'est l'algorithme 3 : les grandes déconnexions consomment beaucoup moins de temps lorsqu'elles sont opérées sur un arbre proche de l'optimum. Par exemple, une seule grande déconnexion de près de 30 turbines est responsable du deux tiers du temps de calcul pour celui à 11,7% de l'optimum (7200 secondes pour une seule déconnexion au tout début). Lorsqu'on impose un temps limite de 30 secondes à cet exemplaire pour résoudre ce problème, le temps chute dramatiquement. Évidemment, à ce moment, les grandes déconnexions ne sont plus nécessairement bien gérées, car leur résolution complète est interrompue, d'où la solution de moins bonne qualité dans le tableau 5.4.

Ces grandes déconnexions seraient les bienvenues donc lors du raffinement d'une solution qui s'est déjà approchée de l'optimum grâce à de plus petites déconnexions.

La deuxième chose à remarquer est qu'il est très envisageable d'utiliser le modèle de Hertz *et al.* (2012) avec les coupes pour les grandes déconnexions. Dans ces cas précis, l'accélération pourrait être substantielle et compenser le temps consommé à générer ces coupes. En effet, la table 5.2 suggère par exemple qu'une connexion de 88 turbines sans coupe peut prendre plus d'une heure, alors que celle avec coupes, 48 secondes... De plus, les grandes déconnexions pourraient être reconnectées avec un temps limite à partir duquel la solution entière trouvée jusqu'à ce moment serait prise comme étant la reconnexion. La majorité du temps d'une grande reconnexion est consommée sur la preuve d'optimalité de la meilleure solution entière que CPLEX trouve.

L'exemplaire 112

L'exemplaire 112 était un autre défi de taille pour le modèle de Hertz *et al.* (2012). Ici, l'algorithme 3 donne une bonne performance, mais il y a toujours une longue durée pour la résolution (433 secondes). Comme à l'exemplaire 60 turbines la majorité du temps est dépensée sur les grandes déconnexions où CPLEX tente de prouver l'optimalité, même s'il trouve très rapidement la solution optimale. Il est intéressant de remarquer que le temps limite n'a pas aidé à réduire le temps de calcul total. C'est parce qu'avec un $\beta = 20$, toutes les déconnexions se font en-dessous de 30 secondes, car il est facile pour CPLEX de reconnecter 20 turbines sur cet exemplaire.

CHAPITRE 6

CONCLUSION

6.1 Synthèse des travaux

Dans ce mémoire, un nouvel algorithme pour la résolution du problème de l'optimisation de la conception d'un réseau de collecte d'énergie éolienne a été développé. Il couple un modèle exact (modèle Hertz *et al.* (2012)) et une heuristique (algorithme 2). Trois remarques importantes ont pu être faites.

1. Bien que le problème de la conception d'un réseau de collecte d'énergie éolienne peut être modélisé comme un problème d'arbre de Steiner, il est important d'observer que, contrairement au modèle classique, la fonction de coût dépend de l'arbre solution, et le problème considéré doit tenir compte de capacités.
2. Étant donnée cette dépendance de la fonction de coût ainsi que les contraintes de capacité, il a été constaté que la méthode classique utilisée pour générer une solution initiale au problème de Steiner (voir Takahashi et Matsuyama (1980)) s'avère inefficace.
3. Enfin, l'idée de Gendreau *et al.* (1998) de déconnecter des noeuds peut être adaptée avec une déconnexion des turbines, et cette idée adaptée s'avère très efficace pour les instances difficiles à résoudre par les modèles exacts. Elle permet de bien diviser le problème en sous-problèmes à résoudre.

6.2 Limitations de la solution proposée

La limitation de la solution proposée réside dans l'algorithme 1 qui échoue parfois à trouver une solution initiale réalisable. Cependant, cette limitation est relative étant donné la solide performance de l'algorithme 3, la lacune serait réparable si un algorithme pouvant générer une solution réalisable était proposé. Même très mauvaise cette solution sera très vite améliorée par l'algorithme 3. Par contre, il reste que l'algorithme 3 est une recherche locale qui, malgré sa large portée lors de grande déconnexion, puisse rester prise dans un optimum local. Idéalement, il faudrait pouvoir générer plusieurs solutions initiales différentes entre elles et leur appliquer l'algorithme 3 indépendamment pour mieux visiter l'espace solution.

6.3 Améliorations futures

Il y a trois améliorations futures qui permettront de rendre le prototype présenté un redoutable engin d'optimisation pouvant s'attaquer à des instances beaucoup plus grosses, des instances dont

les modèles exacts seraient incapables de trouver ne serait-ce qu'une solution réalisable.

1. Connecter itérativement des groupes de turbines dans l'algorithme 1 au lieu d'une turbine à la fois. Ensuite réparer en déconnectant les turbines problématiques, ainsi que des turbines prises aléatoirement.
2. Les coupes suggérées dans Hertz *et al.* (2012) devraient être ajoutées lors des grandes déconnexions pour accélérer la reconnexion. Parallèlement, une résolution d'une déconnexion par Hertz *et al.* (2012) prenant trop de temps à prouver l'optimalité devrait être interrompue lorsque le gap d'optimalité est faible et pas seulement avec un temps limite fixé *a priori* comme il a été fait. Ici, la méthode est heuristique, donc rien ne sert d'attendre la preuve d'optimalité si la connexion trouvée jusqu'à présent améliore la solution courante.
3. La reconnexion d'une déconnexion est simplement la résolution du problème de la conception dans un graphe qui comporte des coûts constants c_{uv}^k différents et des capacités C_{uv} différentes. C'est ce qui donne à la résolution proposée la propriété de *téléscopie*, c'est-à-dire que si la résolution d'une déconnexion trop importante prend trop de temps, il est possible d'appliquer l'algorithme 3 sur l'arbre solution intermédiaire de reconnexion qui est recherché pour trouver une solution proche d'une reconnexion optimale.

Étant donné qu'environ 90% du temps de résolution est déterminé par les grandes déconnexions, plus particulièrement à leurs preuves d'optimalité, ces améliorations devraient réduire substantiellement le temps de calcul.

RÉFÉRENCES

- BERZAN, C., VEERAMACHANENI, K., MCDERMOTT, J. et O'REILLY, U. (2011). Algorithms for cable network design on large-scale wind farms. Rapport technique, Technical Report, Massachusetts Institute of Technology.
- CHAKRAVERTY, S., BATRA, A. et RATHI, A. (2006). Directed convergence heuristic : A fast novel approach to Steiner tree construction. *Very Large Scale Integration, 2006 IFIP International Conference on*. 255 –260.
- DIJKSTRA, E. (1959). A note on two problems in connection with graphs. *Numerische matematik*, 1, 269–271.
- ESBENSEN, H. (1994). Computing near-optimal solutions to the Steiner problem in a graph using a genetic algorithm. *DAIMI PB*, 23.
- FAGERFJÄLL, P. (2010). Optimizing wind farm layout—more bang for the buck using mixed integer linear programming. *Chalmers University of Technology and Gothenburg University*.
- GENDREAU, M., LAROCHELLE, J. et SANZO, B. (1998). *A tabu search heuristic for the Steiner tree problem*. Centre for Research on Transportation - Centre de recherche sur les transports (CRT).
- GLOVER, F. et LAGUNA, M. (1998). *Tabu search*, vol. 1. Springer.
- HERTZ, A., MARCOTTE, O., MDIMAGH, A., CARREAU, M. et WELT, F. (2012). Optimizing the Design of a Wind Farm Collection Network. *INFORMS*, 50, pp. 95–104.
- HWANG, F. et RICHARDS, D. (1992). Steiner tree problems. *Networks*, 22, 55–89.
- KAPSALIS, A., RAYWARD-SMITH, V. et SMITH, G. (1993). Solving the graphical Steiner tree problem using genetic algorithms. *Journal of the Operational Research Society*, 397–406.
- KARP, R. (2010). Reducibility among combinatorial problems. *50 Years of Integer Programming 1958-2008*, 219–241.
- KHOURY, B. et PARDALOS, P. (1996). A heuristic for the Steiner problem in graphs. *Computational Optimization and Applications*, 6, 5–14.
- MACULAN, N. (1987). The steiner problem in graphs. *Annals of Discrete Mathematics*, 31, 185–212.
- RIBEIRO, C. C. et DE SOUZA, M. C. (2000). Tabu search for the Steiner problem in graphs. *Networks*, 36, 138–146.
- TAKAHASHI, H. et MATSUYAMA, A. (1980). An approximate solution for the Steiner problem in graphs. *Math. Japonica*, 24, 573–577.